高职高专计算机教学改革 新体系 规划教材

动态网站设计与制作——JSP

孙 茜 主 编
姚 策 李 娜 副主编

清华大学出版社

高职高专计算机教学改革新体系规划教材

动态网站设计与制作——JSP

孙 茜 姚 策 李 娜 主编

清华大学出版社
北 京

内 容 简 介

JSP(Java Server Pages)是一种动态网页技术标准，利用这一技术可以快速构建跨平台的、先进的、安全的动态网络。本书内容围绕“利用 JSP 进行动态网页制作”这一主题展开，循序渐进地介绍了使用 JSP 进行动态网页制作的相关知识。共分为五章，主要内容为 JSP 概述、Web 开发基础、JSP 语法、JSP 内置对象、JSP 数据库应用等。本书采用案例式教学方法进行编写，以实例应用为主，理论够用为度。书中每一个知识点的结构模式为“案例的提出—关键分析—源代码讲解—相关知识点归纳总结”。

本书适合作为高职院校计算机及相关专业学生学习 JSP 编程的教材，也可供各类培训人员、电子商务人员和程序设计爱好者参考使用。

图书在版编目(CIP)数据

动态网站设计与制作：JSP/孙茜主编. --北京：清华大学出版社，2015
高职高专计算机教学改革新体系规划教材
ISBN 978-7-302-40703-4

Ⅰ. ①动… Ⅱ. ①孙… Ⅲ. ①JAVA 语言－网页制作工具－高等职业教育－教材 Ⅳ. ①TP312 ②TP393.092

中国版本图书馆 CIP 数据核字(2015)第 161245 号

责任编辑：刘士平
封面设计：
责任校对：李 梅
责任印制：

出版发行：清华大学出版社
网　　址：http://www.tup.com.cn，http://www.wqbook.com
地　　址：北京清华大学学研大厦 A 座　　**邮　　编**：100084
社 总 机：010-62770175　　**邮　　购**：010-62786544
投稿与读者服务：010-62776969，c-service@tup.tsinghua.edu.cn
质量反馈：010-62772015，zhiliang@tup.tsinghua.edu.cn
课件下载：http://www.tup.com.cn，010-62795764
印 刷 者：
装 订 者：
经　　销：全国新华书店
开　　本：185mm×260mm　　**印　　张**：15.75　　**字　　数**：359 千字
版　　次：2015 年 8 月第 1 版　　**印　　次**：2015 年 8 月第1 次印刷
印　　数：1～ 000
定　　价： .00 元

产品编号：060266-01

前 言

FOREWORD

动态网站设计与制作——JSP课程是信息专业、网络专业、物联网专业、电子商务专业、云技术专业的一门重要的技术课程。这门课程与Java课程、静态网站设计课程、数据库等也有着密切的联系。

从目前的职业岗位需求来看，网站开发工程师、Java开发工程师、系统维护工程师、数据库工程师、Web工程师、智能硬件云平台系统架构师、云平台软件开发工程师、电子商务平台开发工程师、系统架构师等职业都需要掌握JSP开发技术。通过该课程的学习使学生能够加深对于动态网站、Web应用编程思想和技术的理解与掌握，为学生将来从事相关工作奠定良好的基础。该课程是培养基于Web开发的主要支撑课程，同时也能培养学生良好的编程规范和职业习惯。

JSP是快速建立跨平台的动态网站的首选方案，因为JSP拥有以下几个方面的技术特点。

1. 适应平台。适合大多数的操作系统平台。

2. 执行效率高。拥有Java编程语言“一次编写，各处运行”的特点。

3. 将内容的生成和显示进行分离。

4. 可重用组件。使用JavaBean技术或EJB组件，来执行应用程序所要求的复杂的处理。

5. 采用标识。标准的JSP标识能够访问和实例化JavaBeans组件。

6. 数据库连接。连接数据库的技术是JDBC，目前大多数数据库系统都带有ODBC驱动程序，所以Java程序能访问诸如Oracle、Sybase、MS SQL Server和MS Access等数据库。

本书所使用的开发环境：JDK＋Tomcat＋IE或其他类型的浏览器。

开发工具：MyEclipse＋Dreamweaver。

目前市面上的大多数教材，大多有两个方面的不足：①理论知识的讲解书面化，对于高职学生或自学的社会人员来说并不容易理解；②示范举例过于简单或者过于艰难。所以本书在编写思想上，以适应高职高专教学改革的需要为目标，以企业需求为导向，充分吸收国外经典教材及国内优秀教材的优点，结合我国高校计算机教育的教学现状，打造立体化精品教材。对于高职学生来说，我们培养的不是能够发明和创造代码的人才，而是能够灵活使用、快速入门的应用型人才。

自2013年本书的相关教学资料已经印刷为校内讲义，经过三年的使用、调整、整理才完成本书的初稿。在内容上，以难度适中的“程序体验”为载体，把理论知识与应用相结合，每个程序体验都有相对应的源代码说明，无论是高职学生还是社会自学人员都能够理解和掌握每个题目的关键点，并且安排了阶段性的拓展项目和课后练习，对于知识的活学活用有很大帮助。

本书在形式上，利用网络等现代化手段实现立体化的资源共享，创建了专门的网站，并提供题库、素材、课件、案例分析等内容，实现教师和学生在更大范围内的教学互动，及时解决教学过程中所遇到的问题。

本书采用案例式教学方法进行编写，以实例应用为主，理论够用为度。教程中每一个知识点的结构模式为“案例的提出—关键分析—源代码讲解—相关知识点归纳总结”。

本书由天津轻工职业技术学院孙茜主编，由姚策和李娜担任副主编，王春媚、张润华、皮琳琳和海运职业学院李强参编。

本书专题网站：http://211.81.40.73/dz/dtwz/index.asp。

因时间有限，本书难免存在疏漏之处，敬请读者批评指正。

编　者

2015年5月

目 录

CONTENTS

情境一 JSP的环境配置与开发工具 …… 1

1.1 JSP 简介 …… 1

1.1.1 JSP 技术特点 …… 1

1.1.2 JSP 的运行原理 …… 3

1.1.3 JSP、JavaBean 和 Java Servlet 的关系 …… 4

1.1.4 JSP、ASP、ASP.NET、PHP 的比较 …… 4

1.2 搭建 JSP 运行环境 …… 5

1.2.1 JDK 的下载与安装 …… 5

1.2.2 JDK 环境变量设置 …… 11

1.2.3 Tomcat 的下载与安装 …… 12

1.2.4 测试 Tomcat …… 15

1.3 JSP 开发工具 MyEclipse …… 17

1.3.1 MyEclipse 的下载与安装 …… 17

1.3.2 在 MyEclipse 中配置 Tomcat 服务器 …… 19

1.3.3 使用 MyEclipse 创建第一个 JSP 程序 …… 24

1.4 设置 Web 服务目录 …… 27

1.4.1 Web 服务目录 …… 27

1.4.2 增加虚拟服务目录 …… 27

本章小结 …… 27

本章习题 …… 28

情境二 Web 开发基础——HTML 应用 …… 29

2.1 常用 HTML 标记 …… 29

实战演练 1——段落标记 …… 29

2.1.1 HTML 文件结构 …… 30

2.1.2 HTML 常用标记 …… 31

实战演练 2——在网页中使用各种文本格式 …… 31

2.1.3 HTML 的主要标记——字体标记 …… 33

2.2 HTML 的表格应用 …… 34

实战演练 1——表格 1 …… 34
2.2.1 定义表格的基本语法 …… 35
2.2.2 单元格内文字布局 …… 35
实战演练 2——表格 2 …… 35
2.2.3 跨多行、多列的单元格 …… 36
2.2.4 嵌套表格 …… 36
2.3 在 HTML 中插入图像和超级链接 …… 37
实战演练——超级链接 …… 37
2.3.1 <img>表单标记 …… 38
2.3.2 <a href>标记 …… 38
本章小结 …… 40
本章习题 …… 40

情境三 JSP 语法基础 …… 41

3.1 JSP 注释、表达式 …… 41
实战演练 1——JSP 页面基本结构体验 …… 41
3.1.1 JSP 页面结构 …… 44
3.1.2 注释 …… 44
实战演练 2——字符串作为表达式输出 …… 45
3.1.3 表达式 …… 46
3.2 JSP 的变量声明、方法声明 …… 46
实战演练 1——判断并计算三角形面积 …… 46
3.2.1 使用声明 …… 49
3.2.2 表单的设计 …… 49
实战演练 2——计算圆面积和周长 …… 50
实战演练 3——可累计的投票 …… 52
3.3 JSP 指令 …… 54
实战演练 1——使用 JSP 指令显示当前时间 …… 54
3.3.1 page 指令 …… 56
3.3.2 include 指令 …… 57
3.3.3 Date 对象 …… 57
实战演练 2——include 指令练习 …… 57
3.4 taglib 指令 …… 59
实战演练——taglib 指令引入页脚的练习 …… 59
taglib 指令 …… 61
3.5 JSP 动作 include、param、forward …… 62
实战演练 1——登录 …… 62
3.5.1 <jsp:include>动作标记 …… 66

3.5.2 <jsp:include>动作标记与 include 指令标记的区别 …… 66
3.5.3 param 动作 …… 66
3.5.4 forward 动作 …… 67
实战演练 2——比较大小 …… 67
3.6 useBean 动作 …… 69
实战演练 1——应用 JavaBean 实现图书库存的录入并显示 …… 69
3.6.1 创建 Bean …… 75
3.6.2 <jsp:useBean>动作标记 …… 75
3.6.3 <jsp:setProperty>动作标记 …… 76
3.6.4 <jsp:getProperty>动作标记 …… 77
实战演练 2——应用 JavaBean 实现管理员登录 …… 77
实战演练 3——应用 JavaBean 实现网站访问计数器 …… 81
本章小结 …… 84
本章习题 …… 84

情境四 JSP 内置对象 …… 92

4.1 使用 request 对象 …… 93
实战演练 1——request 计算器 …… 93
request 对象 …… 96
实战演练 2——request 选择题 …… 96
4.2 response 对象 …… 99
实战演练 1——使用 response 实现重定向 …… 99
4.2.1 out 对象 …… 102
4.2.2 response 对象 …… 102
实战演练 2——网页保存为 Word 文档 …… 103
实战演练 3——网页定时刷新 …… 105
4.3 session 对象 …… 105
实战演练——简易购物 …… 105
4.3.1 session 对象 …… 108
4.3.2 session 对象常用方法 …… 108
4.4 application 对象 …… 109
实战演练——使用 application 制作站点计数器 …… 109
4.4.1 application 对象 …… 110
4.4.2 application 对象常用方法 …… 111
4.5 综合实战演练：购物车 …… 111
本章小结 …… 117
本章习题 …… 117

情境五　JSP数据库应用 …… 121
5.1　SQL数据库的安装 …… 121
5.1.1　SQL Server 2000的安装 …… 121
5.1.2　SQL Server 2000的使用 …… 126
5.2　使用JDBC—ODBC技术连接数据库 …… 131
实战演练——用户的注册登录 …… 131
5.2.1　JDBC技术和驱动程序 …… 143
5.2.2　JSP中使用JDBC—ODBC访问数据库 …… 143
5.3　JSP对数据库的操作 …… 144
实战演练1——诗词目录 …… 144
实战演练2——图书库存情况的查找 …… 153
实战演练3——留言板 …… 158
本章小结 …… 167
本章习题 …… 167
附录1　源代码 …… 176
附录2　知识字典 …… 221
附录3　习题集锦 …… 230
参考文献 …… 241

情境一

JSP 的环境配置与开发工具

Project 1

1.1 JSP 简 介

JSP(Java Server Pages)是由 Sun Microsystems 公司倡导、许多公司参与一起建立的一种动态网页技术标准。JSP 技术类似于 ASP 技术,它是在传统的网页 HTML(标准通用标记语言的子集)文件(*.htm,*.html)中插入 Java 程序段(Scriptlet)和 JSP 标记(tag),从而形成 JSP 文件(*.jsp)。用 JSP 开发的 Web 应用是跨平台的,既能在 Linux 上运行,也能在其他操作系统上运行。

动态网页和静态网页的根本区别在于服务器端返回的 HTML 文件是事先存储好的还是由程序即时生成的。静态网页文件里只有 HTML 标记,没有程序代码,网页的内容都是事先写好并存放在服务器上的;动态网页文件不仅含有 HTML 标记,而且还含有程序代码,当用户发出请求时,服务器由动态网页程序即时生成 HTML 文件。动态网页能够根据不同的时间、不同的用户生成不同的 HTML 文件,显示不同的内容。

1.1.1 JSP 技术特点

JSP 是快速建立跨平台的动态网站的首选方案,因为 JSP 拥有以下几个技术特点。

1. 适应平台

由于 JSP 的脚本语言是 Java 语言,因此它具有 Java 语言的一切特性。几乎所有平台都支持 Java,所以 JSP+JavaBean 几乎可以在所有平台通行无阻。从一个平台移植到另一个平台,JSP 和 JavaBean 甚至不用重新编译,因为 Java 字节码都是标准化的,与平台无关。

2. 执行效率高

由于 JSP 页面的内置脚本语言是基于 Java 的,而且所有的 JSP 页面都被编译成为 Java Servlets,所以 JSP 页面具有 Java 技术的所有优点,包括健壮的存储管理和安全性。作为 Java 平台的一部分,JSP 拥有 Java 编程语言“一次编写,各处运行”的特点。

当JSP第一次被请求时,JSP页面转换成Servlet,然后被编译成.class文件,以后(除非页面有改动或Web服务器被重新启动)再有客户请求该JSP页面时,JSP页面不需重新编译,而是直接执行已编译好的.class文件,因此执行效率高。

3. 将内容的生成和显示进行分离

用JSP技术,Web页面开发人员可以使用HTML或者XML标识来设计和格式化最终页面。生成内容的逻辑被封装在标识和JavaBean组件中,并且捆绑在脚本中,所有的脚本在服务器端运行。由于核心逻辑被封装在标识和JavaBean中,所以,Web管理人员和页面设计者能够编辑和使用JSP页面,而不影响内容的生成。

在服务器端,JSP引擎解释JSP标识和脚本,生成所请求的内容,并且将结果以HTML(或者XML)页面的形式发送回浏览器。这既有助于作者保护自己的代码,又能保证任何基于HTML的Web浏览器的完全可用性。

4. 可重用组件

绝大多数JSP页面依赖于可重用的、跨平台的JavaBean或EJB(Enterprise JavaBean)组件来执行应用程序所要求的复杂的处理,为JSP程序的开发提供了方便。例如,用户可以将复杂的处理程序或需要重复利用的程序(如对数据库的操作)封装到组件中,在开发中可以多次使用这些组件,提高了组件的重用性。基于组件的方法加速了总体开发过程,并且使得各种组织在他们现有的技能和优化结果的开发努力中得到平衡。

5. 采用标识

Web页面开发人员不会都是熟悉脚本语言的编程人员。JSP技术封装了许多功能,这些功能是在易用的、与JSP相关的XML标识中进行动态内容生成所需要的。标准的JSP标识能够访问和实例化JavaBean组件,设置或者检索组件属性,下载Applet,以及执行用其他方法更难于编码和耗时的功能。

6. 数据库连接

Java中连接数据库的技术是JDBC,Java程序通过JDBC驱动程序与数据库相连,执行查询、提取数据等操作。Sun公司还开发了JDBC—ODBC bridge,利用此技术,Java程序可以访问带有ODBC驱动程序的数据库,目前大多数数据库系统都带有ODBC驱动程序,所以Java程序能访问诸如Oracle、Sybase、MS SQL Server和MS Access等数据库。

此外,通过开发标识库,JSP技术还可以进一步扩展。第三方开发人员和其他人员可以为常用功能创建自己的标识库。这使得Web页面开发人员能够使用熟悉的工具和执行特定功能的构件来进行工作。

JSP技术很容易整合到多种应用体系结构中,以利用现存的工具和技巧,并且能扩展到支持企业级的分布式应用中。作为采用Java技术家族的一部分,以及Java 2(企业版体系结构)的一个组成部分,JSP技术能够支持高度复杂的基于Web的应用。

1.1.2 JSP的运行原理

在 JSP 页面中，除了普通的 HTML 代码，还有以下 3 种重要的组成部分：脚本元素、指令和动作。脚本元素用来使 Java 代码可以直接插入 Servlet；JSP 指令用来从整体上控制 Servlet 的结构；动作用来引入现有的组件或控制 JSP 引擎的行为。比较常见的 JSP 引擎是 Tomcat，把支持 JSP 的 Web 服务器配置好以后，就可以在客户端通过浏览器来访问 JSP 页面了。

如图 1-1 所示是 JSP 技术生成动态页面内容的过程。

一个 JSP 页面的执行过程可以分为以下几个步骤。

（1）浏览器客户端向 JSP 引擎发出对 JSP 页面的请求。

（2）JSP 引擎将对应的 JSP 页面转换成 Java Servlet 源代码。

（3）JSP 引擎编译生成源代码，生成.class 文件。

（4）JSP 引擎加载运行对应的.class 文件，生成响应的结果页面(JVM 是 java 虚拟机，将 java 的 class 字节码文件翻译成二进制可执行程序)。

（5）JSP 引擎把响应的输出结果发送到客户端浏览器。

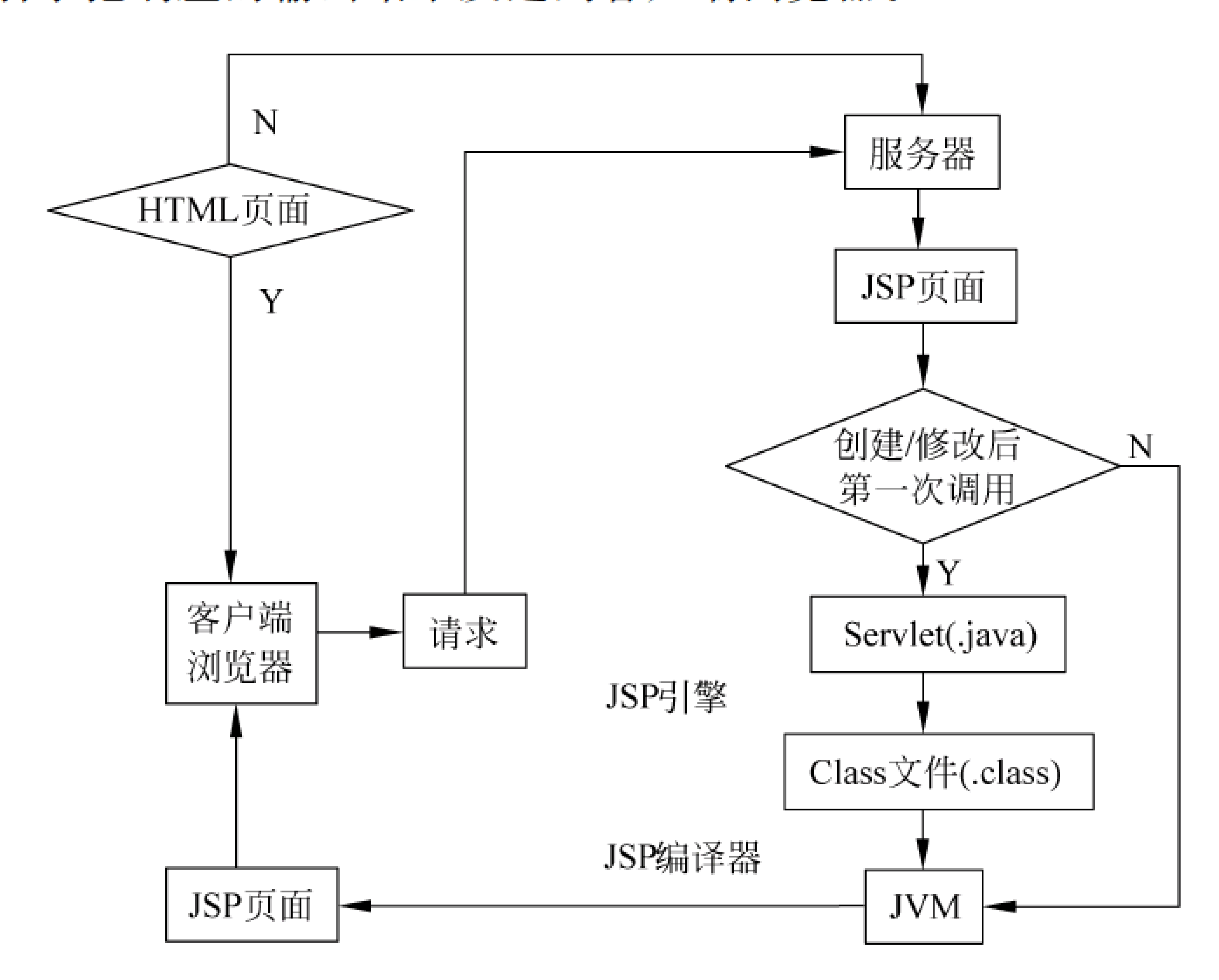


图 1-1 JSP 工作流程

JSP 引擎管理 JSP 页面生命周期的两个阶段：转换阶段和执行阶段。当有一个对 JSP 页面的请求到来时，JSP 引擎检验 JSP 页面的语法是否正确，将 JSP 页面转换为 Servlet 源文件，然后调用 Javac 工具类编译 Servlet 源文件生成字节码文件，这一阶段是转换阶段。接下来，Servlet 引擎加载转换后的 Servlet 类，实例化一个对象处理客户端的请求，在请求处理完成后，响应对象被 JSP 引擎吸收，引擎将 HTML 格式的响应信息发送到客户端，这一阶段是执行阶段。

从整个过程中可以知道，当第一次加载 JSP 页面时，因为要将 JSP 页面转换为 Servlet 源文件，所以响应速度较慢。当再次请求时，JSP 引擎就会直接执行第一次请求

时产生的 Servlet，而不会再重新转换 JSP 页面，所以其执行速度和原始的 Servlet 执行速度几乎就一样了。在 JSP 执行期间，JSP 引擎会查看 JSP 页面，看是否有更新或修改。如果有更新或修改，JSP 引擎会再次编译 JSP 或 Servlet；如果没有更新或修改，就直接执行前面产生的 Servlet。

1.1.3 JSP、JavaBean 和 Java Servlet 的关系

Java Servlet 是 Java 语言的一部分，它提供了一组用于服务器端编程的 API。习惯上称使用 Java Servlet API 的相关类和方法所编写的 Java 类为 Servlet 类，Servlet 类生成的对象为 Servlet 对象。Servlet 对象可以在配置有 JSP 运行环境的服务器上运行，访问服务器的各种资源，这极大地扩展了服务器的功能。

JSP 是晚于 Java Servlet 产生的以 Java Servlet 技术为基础的 Web 应用开发技术标准。JSP 保留了 Java Servlet 几乎所有的优点，是 Java Servlet 技术的成功应用，但 JSP 只是 Java Servlet 技术的一部分，而不是 Java Servlet 的全部。JSP 可以让 JSP 标记、Java 语言代码嵌入 HTML 语句中，大大地简化和方便了网页的设计和修改，但 JSP 页面最终会被编译成 Servlet 并执行，来响应客户端的请求。

JavaBean 被 Sun 公司定义为一个可重用的软件组件。实际上，JavaBean 就是一种 Java 类，通过封装属性和方法成为具行某种业务逻辑处理能力的类，它一般负责 Web 应用系统的业务逻辑处理部分。JavaBean 类实例化的对象简称为 Bean。JSP 提供访问 JavaBean 组件的 JSP 动作标记。JSP 动作标记简单、方便，有效地分离了 JSP 页面的表示部分和业务逻辑、数据处理部分，因此使程序设计人员和页面设计人员可以同时工作。

较小规模的 Web 应用可以采用 JSP＋JavaBean 模式。在 JSP＋JavaBean 模式中，JSP 负责页面的显示、页面预处理和跳转控制，JavaBean 负责业务逻辑和数据处理。对于规模较大的 Web 应用，就需要采用 JSP＋JavaBean＋JavaServlet 模式。在 JSP＋JavaBean＋JavaServlet 模式中，JSP 负责页面显示（View），JavaBean 负责业务逻辑和数据处理（Model），JavaServlet 负责处理和分发页面的请求（Controler）。

1.1.4 JSP、ASP、ASP.NET、PHP 的比较

JSP 是一种动态网页技术标准，主要是 Java 语言在 Web 方面的运用。JSP 的优势是跨平台，既能在 Linux 下运行，也能在其他操作系统上运行。经过这些年的发展，JSP 技术在市场上的应用越来越广泛，而且一般为大中型项目，这就带来了更多的就业机会。由于 Java 发展经久不衰，所以 JSP 技术的应用也有了更广阔的空间。对于学习 Java 的同学们，JSP 也是必学的技术之一，许多培训机构也会将 JSP 作为 Java 的基础课程讲授。

对于做网站，也就是 Web 类服务，曾经特别流行三大体系，即 JSP、ASP 和 PHP。后来经过微软对 ASP 的升级换代，目前 ASP 基本淡出舞台了，取而代之的是 ASP. NET，ASP 可以说是 ASP. NET 的前身。实际上，JSP、ASP 和 PHP 都只是一种技术，本身不是语言。

那么，JSP、ASP 和 PHP，到底有什么区别，学习哪个更有前途？今天就为大家分析一下。

目前，PHP 与 ASP 在国内应用最为广泛，而 JSP 采用的较少。但在国外，JSP 已经是比较流行的一种技术，尤其是电子商务类的网站多采用 JSP。

采用 PHP 的网站如新浪网(sina)、中国人(Chinaren)等，但 PHP 不适合应用于大型的电子商务网站，而更适合一些小型的商业网站，因为 PHP 缺乏规模支持和多层结构支持。对于大负荷站点，解决方法只有一个：分布计算。数据库、应用逻辑层、表示逻辑层彼此分开，而且同层也可以根据流量分开，组成二维阵列，而 PHP 则缺乏这种支持。JSP 可以利用 JDBC 和众多主流关系数据库连接，PHP 目前只支持 MySql 和 Oracle 等有限的几种，此外 PHP 提供的数据库接口支持不统一，这也使得它不适合运用在电子商务中。

ASP 是 Active Server Pages 的缩写，是微软在早期推出的动态网页制作技术，包含在 IIS(Internet 信息服务)中，是一种服务器端的脚本编写环境，可以用于创建和运行动态、交互的 Web 服务器应用程序。在动态网页技术发展的早期，ASP 是绝对的主流技术，但是它也存在着许多缺陷：由于 ASP 的核心是脚本语言，所以需要通过解释执行代码，因此运行效率较低；同时由于脚本代码与 HTML 代码混在一起，因此不便于开发人员进行管理与维护。随着技术的发展，ASP 的辉煌已经成为过去，微软也已经不再对 ASP 提供技术支持和更新，ASP 技术目前处于被淘汰的边缘。

ASP. NET 是微软继 ASP 后推出的全新的动态网页制作技术。在性能上，ASP. NET 比 ASP 强得多，与 PHP 相比，也存在显著优势。ASP. NET 可以使用 C#、VB. NET、Visual J# 等语言来开发。ASP. NET 依托. NET 平台先进而强大的功能，极大地简化了编程人员的工作，使得 Web 应用程序的开发更加方便、快捷，同时也使得程序的功能更加强大，是 JSP 技术的有力竞争对手。然后在 ASP. NET 代替 ASP 之后的这些年，C# 还不足以与 Java 竞争。对于从业人员的待遇更是不如 Java。所以，ASP. NET 的前景未卜。

1.2　搭建 JSP 运行环境

使用 JSP 开发 Web 应用程序时，必须搭建一个 JSP 运行环境。JSP 运行环境至少需要具备三个基本条件：要在用户的计算机上安装 Java 的 JDK(Java Development Kit)，并进行环境变量的设置；要在计算机上安装 JSP 引擎，例如 Tomcat；要在计算机上有浏览器。

1.2.1　JDK 的下载与安装

JDK 是 Sun 公司针对 Java 开发员所设计的产品。JDK 是整个 Java 的核心，包括了 Java 运行环境、Java 工具和 Java 基础的类库。

JSK 的版本分类如下。

(1) SE(J2SE)，standard edition，标准版，是通常使用的一个版本，从 JDK 5.0 开始，改名为 Java SE。

(2) EE(J2EE)，enterprise edition，企业版，使用这种 JDK 可开发 J2EE 应用程序，从

JDK 5.0 开始,改名为 Java EE。

(3) ME(J2ME),micro edtion,主要用于移动设备、嵌入式设备上的 Java 应用程序开发,从 JDK 5.0 开始,改名为 Java ME。

1. JDK 的下载

(1) 首先,我们可以在 Sun 公司的官方网站 http://java.sun.com 下载 JDK 的最新版本。进入官网后我们会看到以下界面,选择 Downloads 菜单——Java SE,如图 1-2 所示。



图 1-2 Sun 公司的官方网站

(2) 进入 Java SE 页面,单击 JDK 6 Update 19 右侧的 Download JDK 按钮,如图 1-3 所示。



图 1-3 Java SE 页面

(3) 进入下载页面,选择操作平台 Windows,勾选上复选框表示同意协议,单击 Download 按钮,如图 1-4 所示。

(4) 弹出登录的窗口如图 1-5 所示,可以单击 Skip this Step 跳过这个步骤,即可进入最后一步。

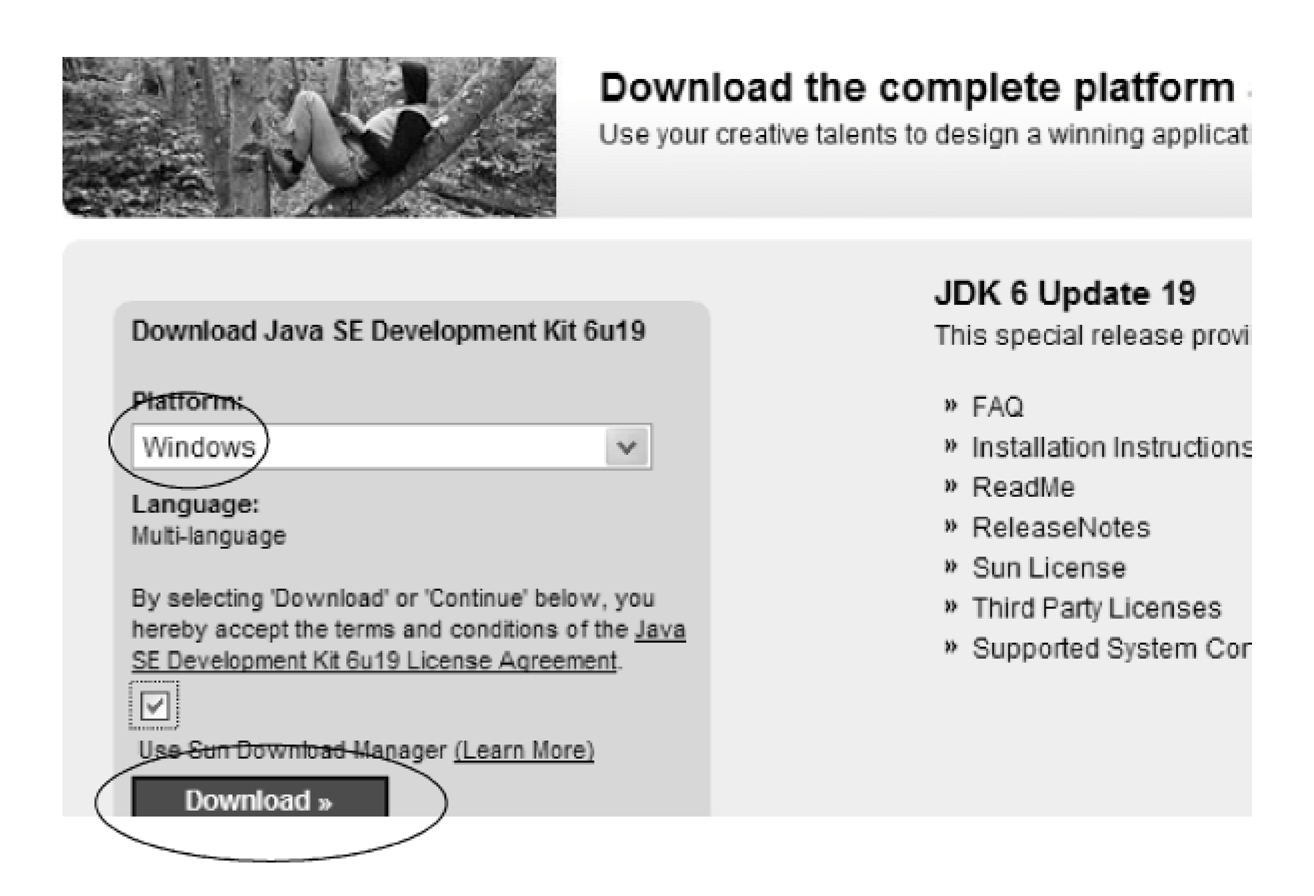


图 1-4　JDK 下载页面

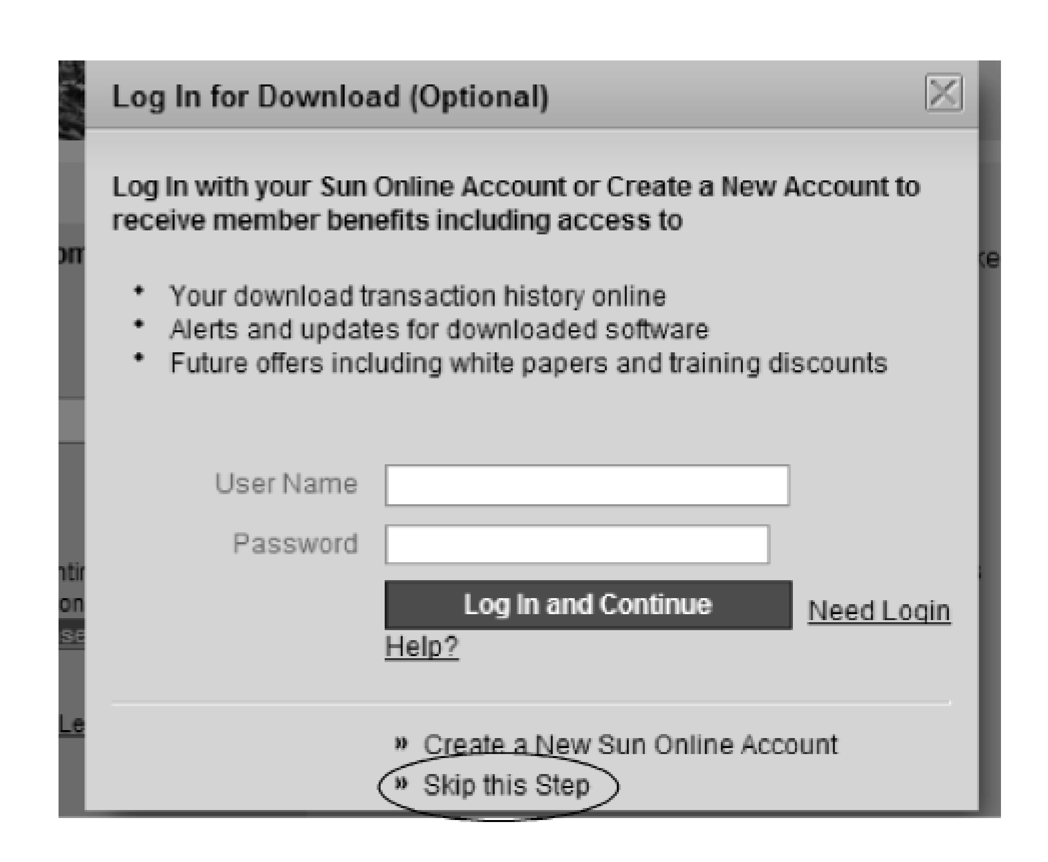


图 1-5　JDK 下载登录页面

（5）直接进入下载页面如图 1-6 所示，单击 jdk-6u19-windows-i586.exe 下载文件。

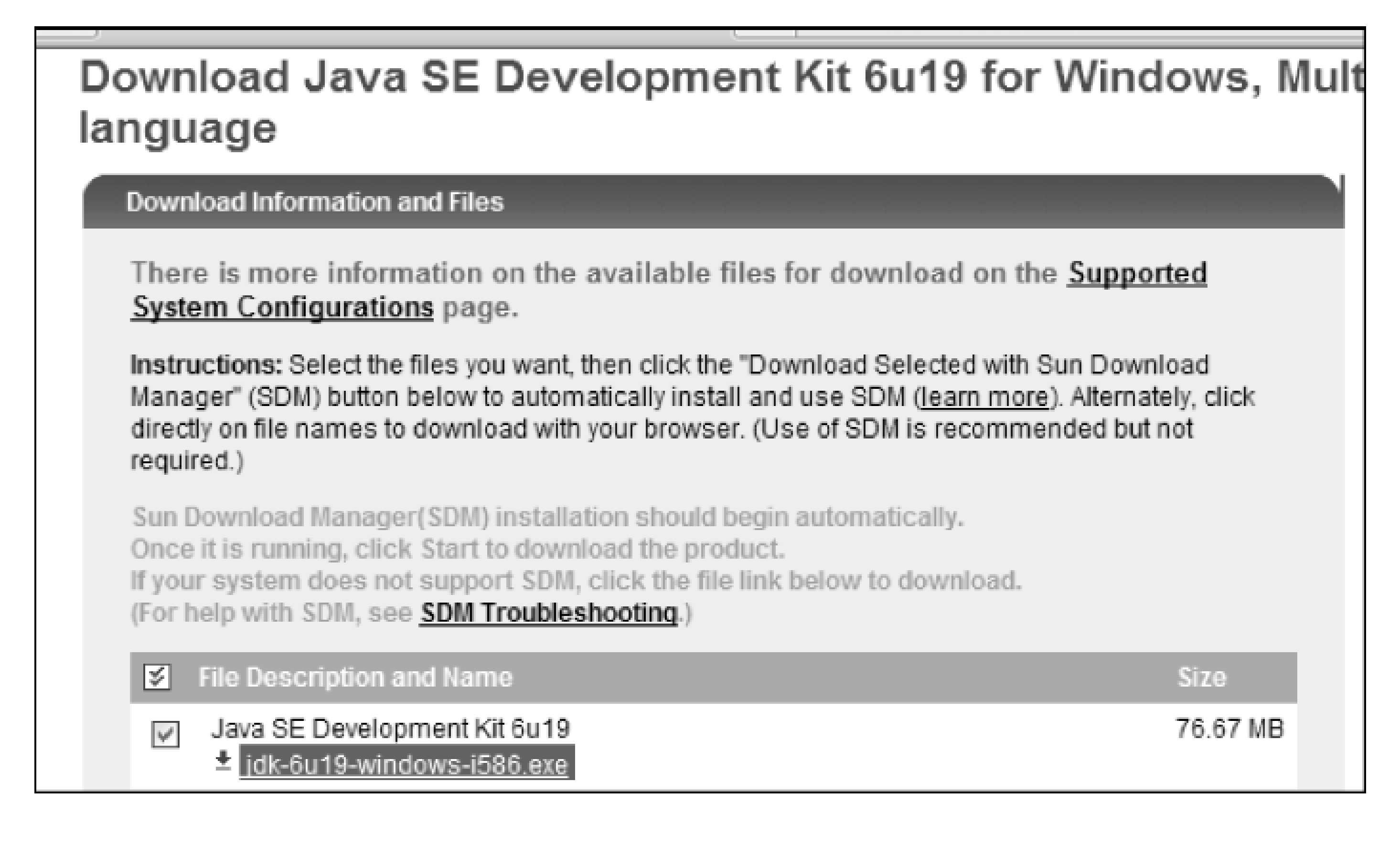
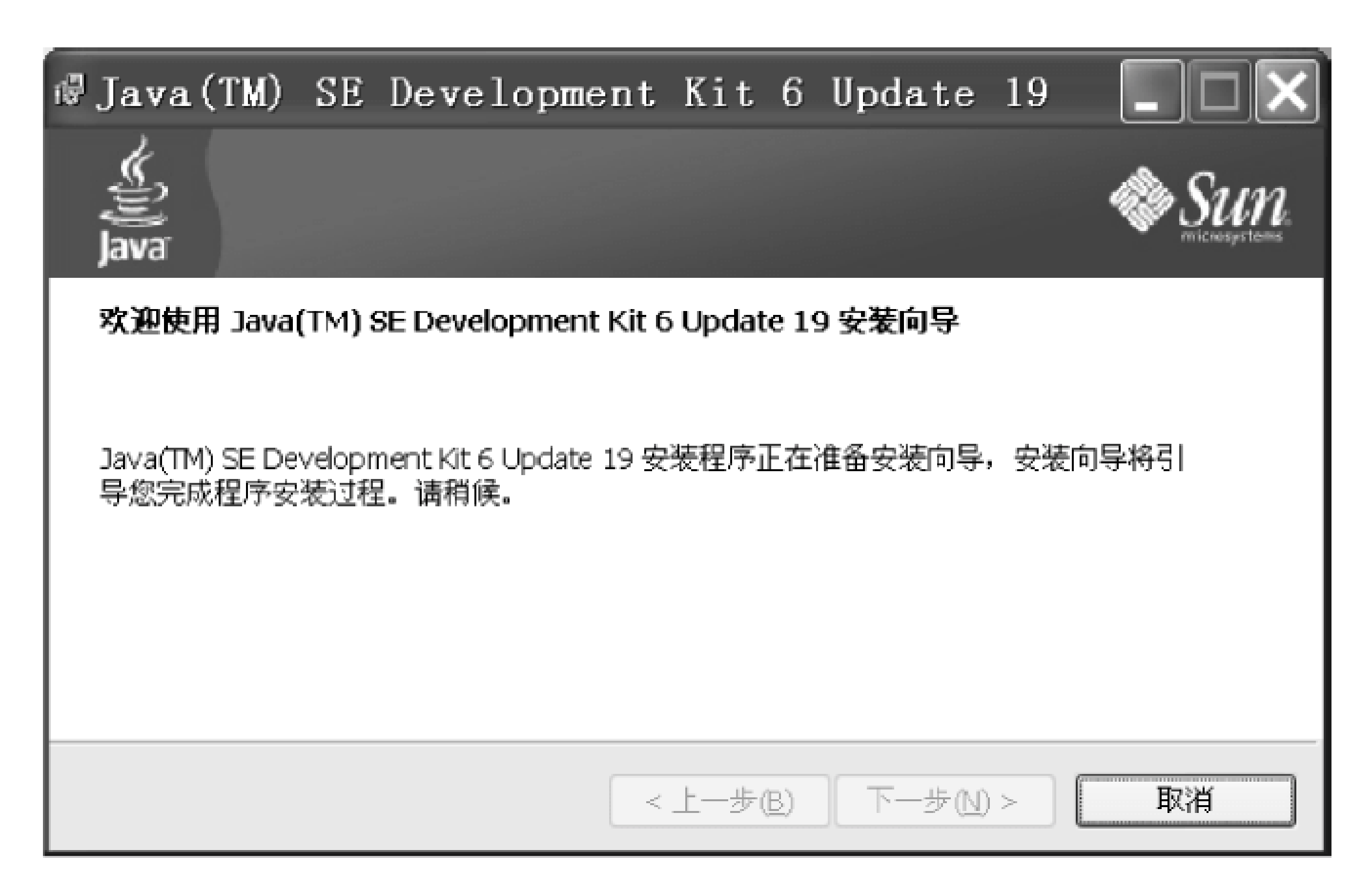


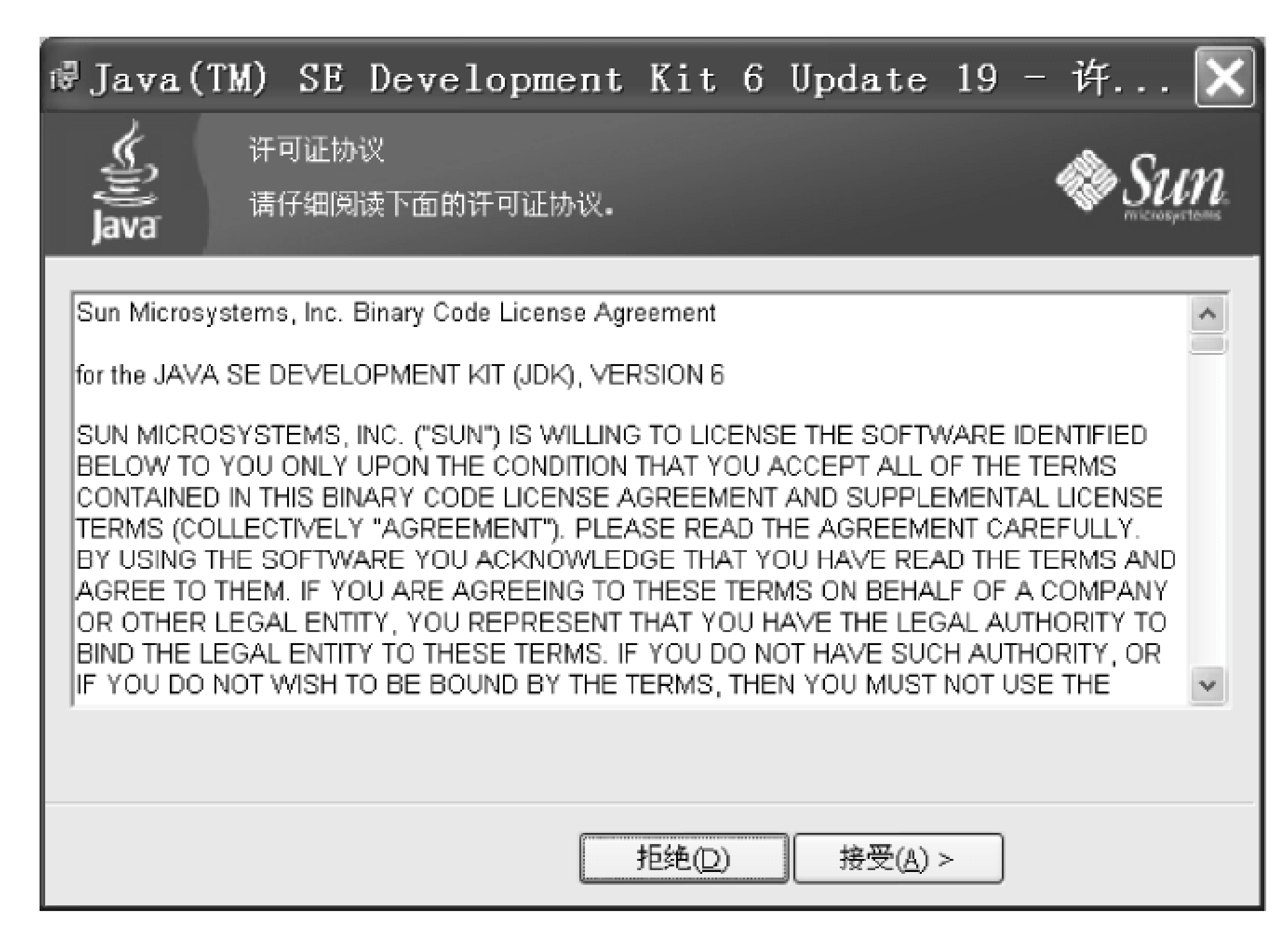
图 1-6　JDK 程序下载页面

2. 安装

（1）双击刚下载好的jdk-6u19-windows-i586.exe，弹出JDK欢迎界面如图1-7(a)所示和协议界面如图1-7(b)所示。



(a) JDK安装欢迎界面



(b) JDK协议界面

图 1-7

（2）单击“接受”按钮将出现如图1-8所示窗口，单击“更改”按钮，修改安装路径如图1-9所示，单击“确定”按钮，确定已修改的路径，然后单击“下一步”按钮，完成JDK的安装。

（3）安装过程中弹出JRE的安装窗口，如图1-10所示，单击“更改”按钮把路径修改为d:\Program Files\Java\jre6\，如图1-11所示，单击“确定”按钮，继续安装。

（4）安装完成弹出窗口，如图1-12所示。



图 1-8　JDK 自定义安装界面



图 1-9　JDK 安装路径选择界面

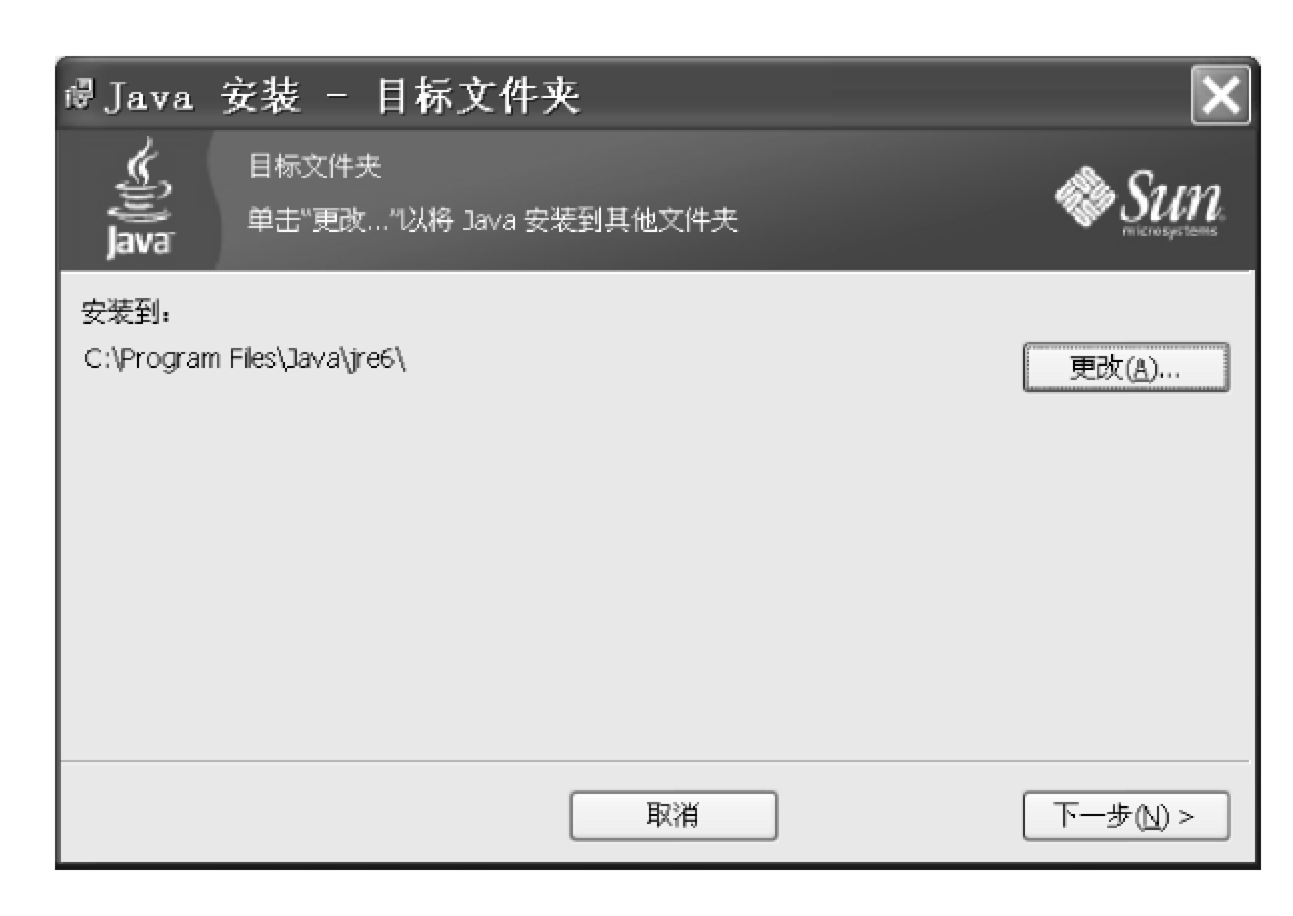


图 1-10　JRE 安装窗口

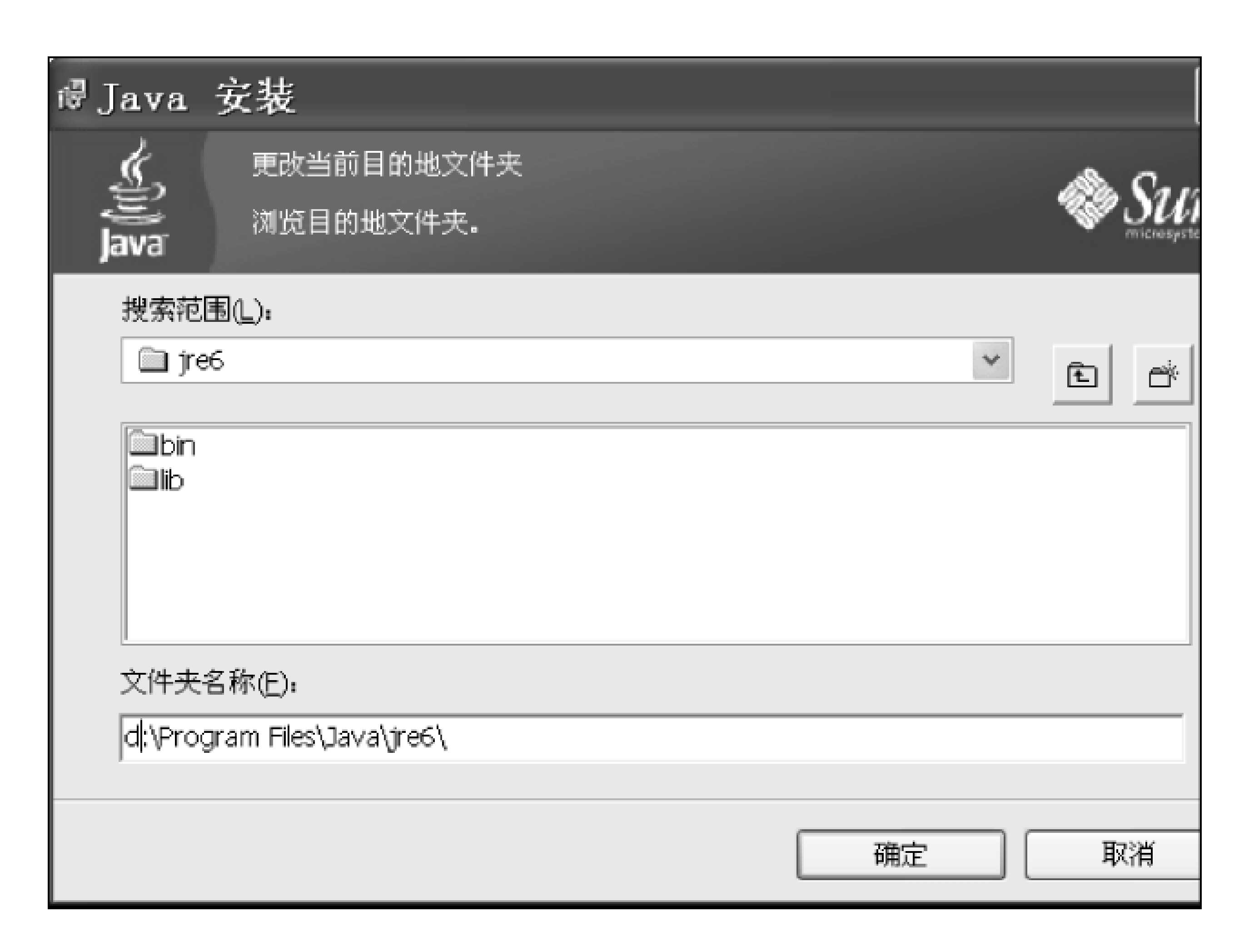


图 1-11　JRE 安装路径选择界面



图 1-12　JDK 安装完成窗口

3．关于 JDK 和 JRE

刚才在安装 JDK 的时候让我们选择是否安装单独的 JRE，一般都会一起安装，因为这样更能帮助大家弄清楚它们的区别。

现在我们来看一下安装好的 java 目录，在目录下有一个 jdk 目录和一个 jre 目录。

JDK 是 Java 的开发工具包，里面包含了各种类库和工具。jdk 目录下面有 6 个文件夹、一个 src 类库源码压缩包和其他几个声明文件。其中，真正在运行 java 时起作用的是以下四个文件夹：bin、include、lib、jre。

bin：编译器(javac.exe)；

include：Java 和 JVM 交互用的头文件；

lib：所有 Java 类库的 class 文件，都在 lib 目录下打包成了 jar；

jre：java runtime environment，是 Java 程序的运行环境。既然是运行，当然要包含 jvm，也就是虚拟机，JDK 的 jre/bin/client 和 server 两个文件夹下都包含一个 jvm. dll，说明是有两个虚拟机的。

jdk 目录下的 bin 文件夹中有各种 Java 程序需要用到的命令，与 jre 目录下的 bin 文件夹最明显的区别在于 jdk 下才有 javac，具备开发功能的 jdk 自己的 jre 下才会同时有 client 性质的 jvm 和 server 性质的 jvm，而仅仅作为运行环境的 jre 下只需要 client 性质的 jvm. dll 就够了(jdk 目录外的 jre 只是 java 的运行环境)。

简单地说，JDK 是面向开发人员使用的，它提供了 Java 的开发环境和运行环境，可以包括函数库、编译程序等。JRE 是 Java 的运行环境，是面向 Java 程序的使用者，而不是开发者。jdk 目录里的工具是用 Java 编写的，所以也是 Java 的应用程序，因此也必须要自行附一套 jre 才行。

1.2.2　JDK 环境变量设置

(1) 右击“我的电脑”—“属性”，在弹出窗口上点击“高级”选项卡，如图 1-13 所示。

(2) 单击“环境变量”按钮，弹出窗口如图 1-14 所示，新建系统变量：

path=D:\Program Files\Java\jdk1. 6. 0_19\bin ，如图 1-15 所示；

classpath=. ;D:\Program Files\Java\jdk1. 6. 0_19\jre\lib，如图 1-16 所示。

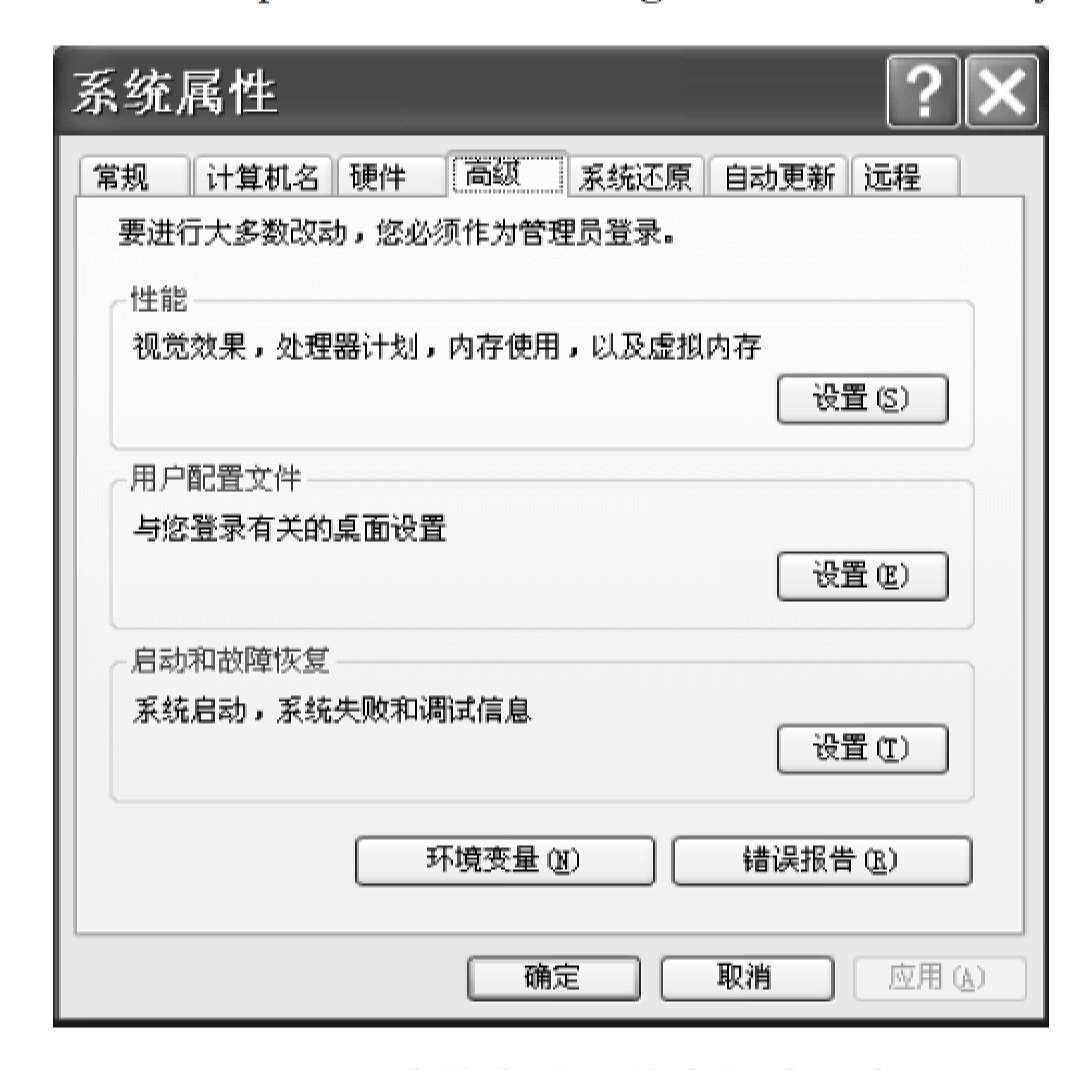


图 1-13　我的电脑属性高级选项卡



图 1-14　环境变量设置窗口



图 1-15　新建系统变量窗口(path)



图 1-16　新建系统变量窗口(classpath)

(3) 打开"开始"—"运行",在弹出窗口输入 cmd。

(4) 在窗口中输入代码如下:

```
d:          //d:表示打开 D 盘
javac       //javac 是 java 程序的编译器
```

单击 Enter 键后显示结果如图 1-17 所示,列出了 JDK 中 javac 的用法,这就表示测试环境成功了。



图 1-17 JDK 环境测试结果窗口

1.2.3 Tomcat 的下载与安装

Tomcat 是 Apache 软件基金会(Apache Software Foundation)的 Jakarta 项目中的一个核心项目,由 Apache、Sun 和其他一些公司及个人共同开发而成。由于有了 Sun 的参与和支持,最新的 Servlet 和 JSP 规范总是能在 Tomcat 中得到体现,Tomcat 5 支持最新的 Servlet 2.4 和 JSP 2.0 规范。因为 Tomcat 技术先进、性能稳定,而且免费,因而深受 Java 爱好者的喜爱,并得到了部分软件开发商的认可,成为目前比较流行的 Web 应用服务器。目前最新版本是 6.0。

Tomcat 是一个小型的轻量级应用服务器,是开发和调试 JSP 程序的首选。对于初学者来说,可以认为,当在一台机器上配置好 Apache 服务器,可利用它响应对 HTML 页面的访问请求。Tomcat 和 IIS、Apache 等 Web 服务器一样,具有处理 HTML 页面的功能,另外它还是一个 Servlet 和 JSP 容器,独立的 Servlet 容器是 Tomcat 的默认模式。实际上,Tomcat 部分是 Apache 服务器的扩展,但它是独立运行的,所以当运行 Tomcat 时,实际上是作为一个与 Apache 独立的进程单独运行的。

1. 下载 Tomcat

(1) 可以从 http://tomcat.apache.org/网站下载,进入官网后我们会看到以下界面,如图 1-18 所示。

(2) 单击 Tomcat 6.x 链接,进入下载页面,单击 Windows Service Installer,就可以下载 Tomcat 6.0 了,如图 1-19 所示。

2. 安装 Tomcat

这里使用的 Tomcat 是 6.0 版本,平台为 Windows,安装之前必须完成上述的 JDK 的安装。

(1) 双击已下载好的 apache-tomcat-6.0.26.exe 文件,将出现如图 1-20 所示的界面。

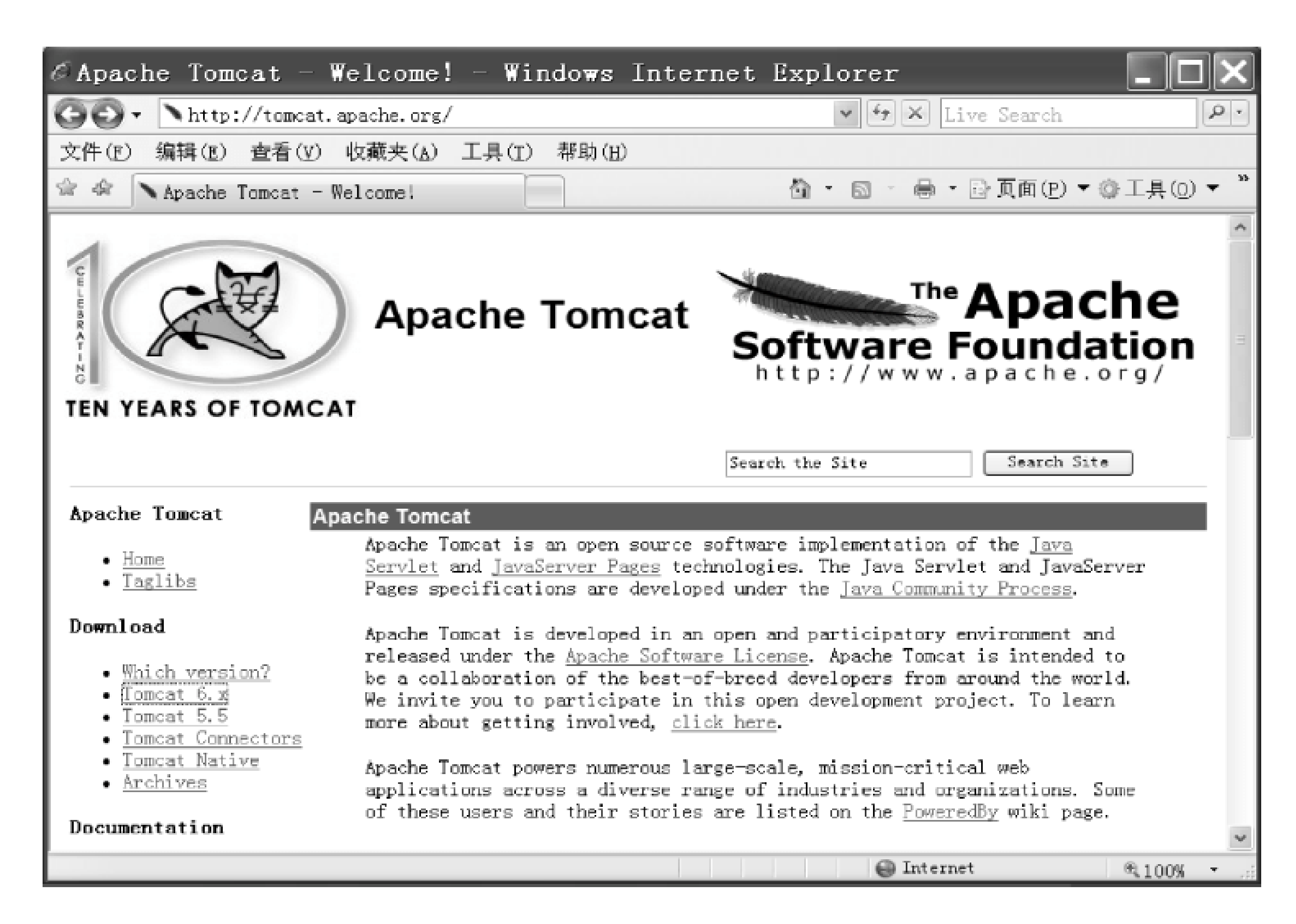


图 1-18　Apache Tomcat 官方网站

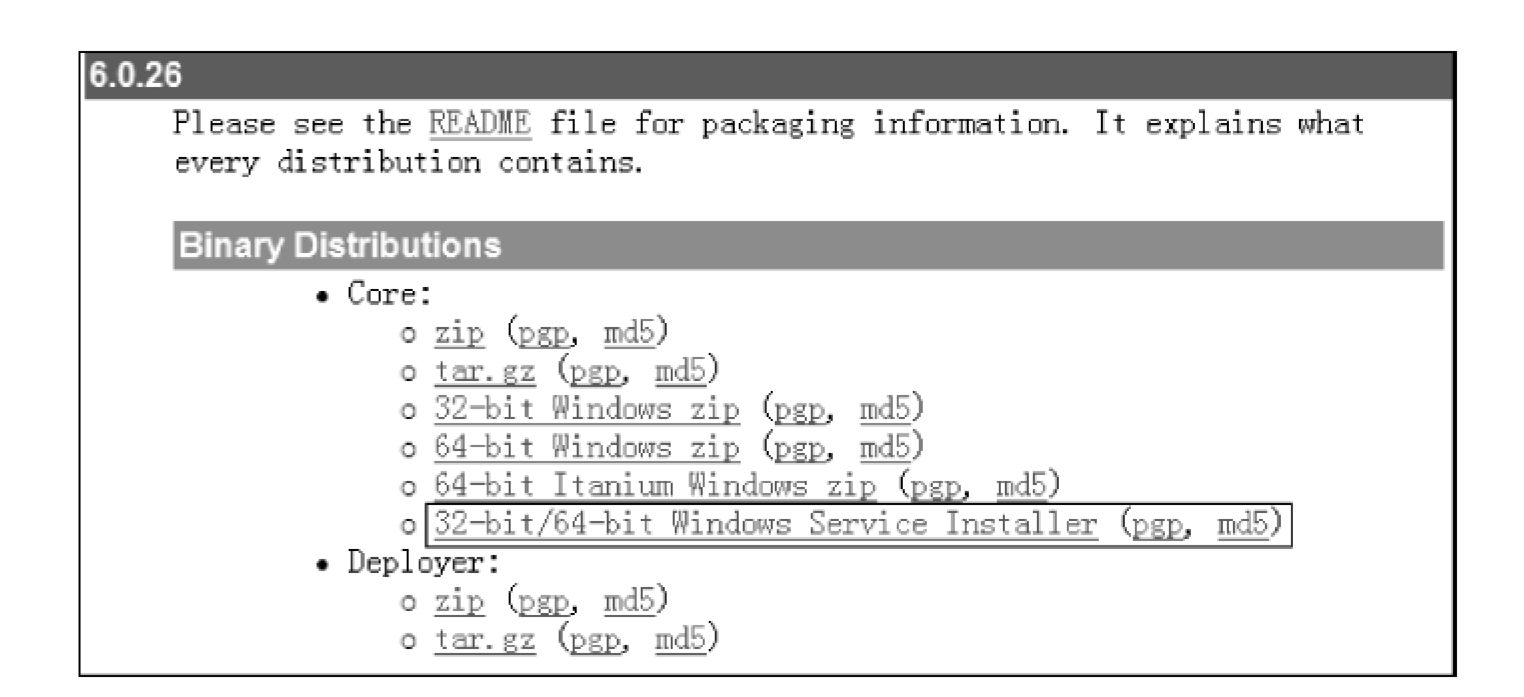


图 1-19　Tomcat 下载界面

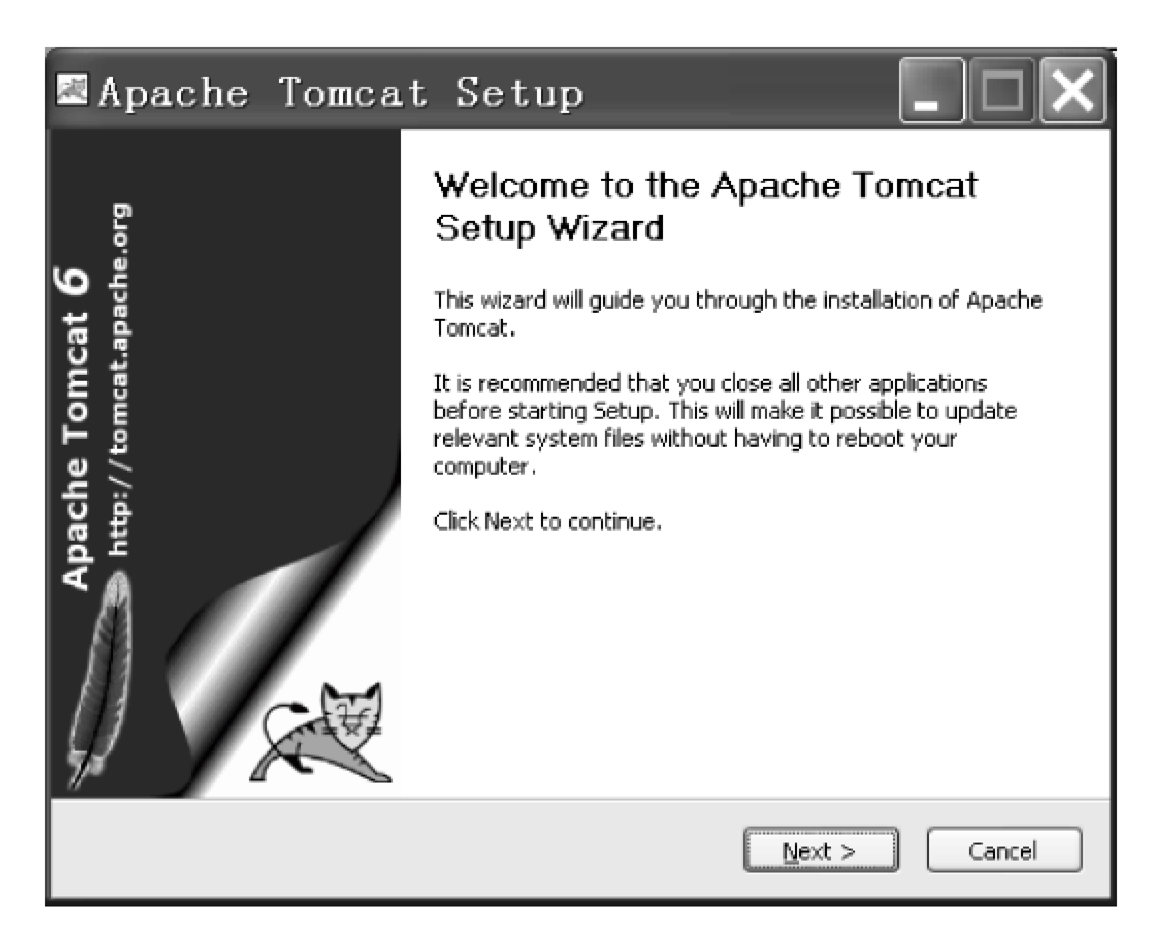


图 1-20　Tomcat 安装界面

（2）单击 Next 按钮，选择下拉菜单中的 Full，如图 1-21 所示。

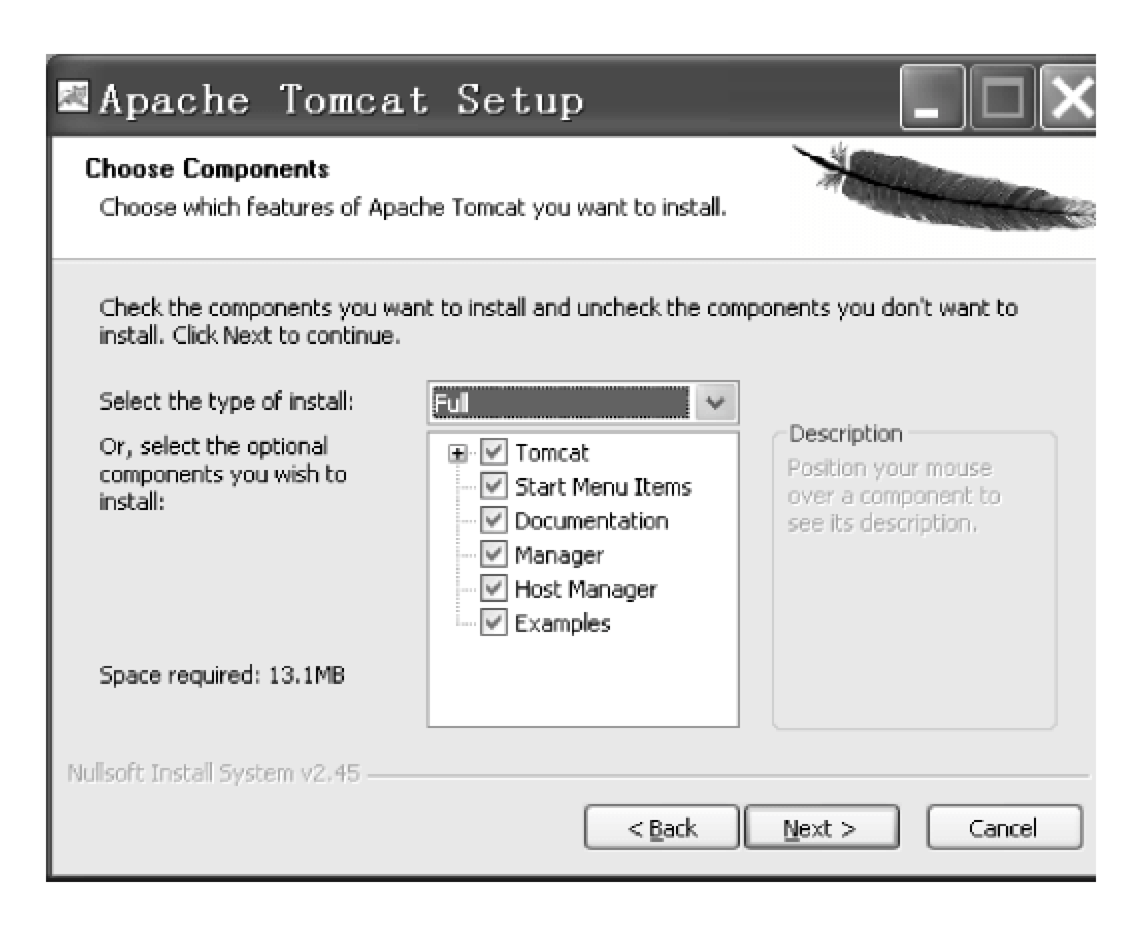


图 1-21 Tomcat 安装选择界面

（3）单击 Browse 修改安装路径如图 1-22 所示，单击 Next 开始安装，在安装过程中会弹出如图 1-23 所示的界面，如果 JDK 安装正确，Tomcat 会自动识别 jre 的路径，单击 Install 按钮，完成安装。

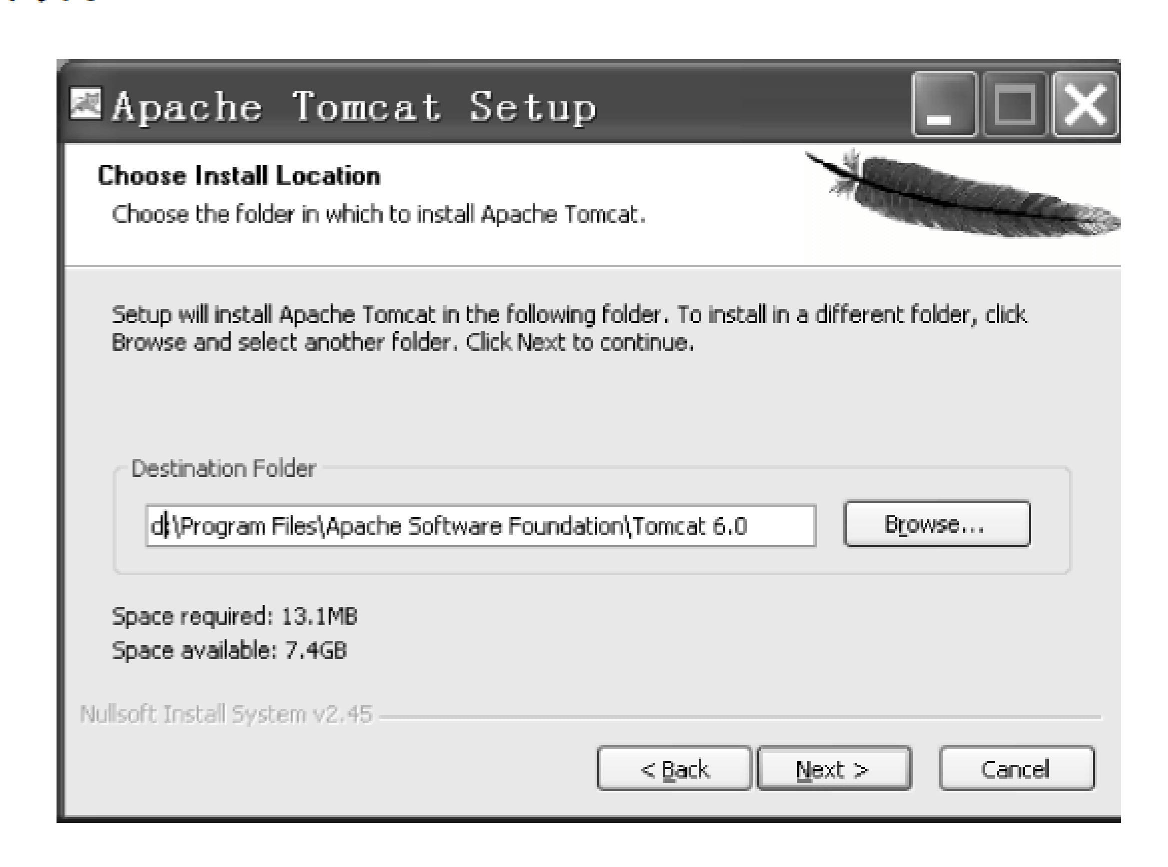


图 1-22 Tomcat 安装路径设置界面

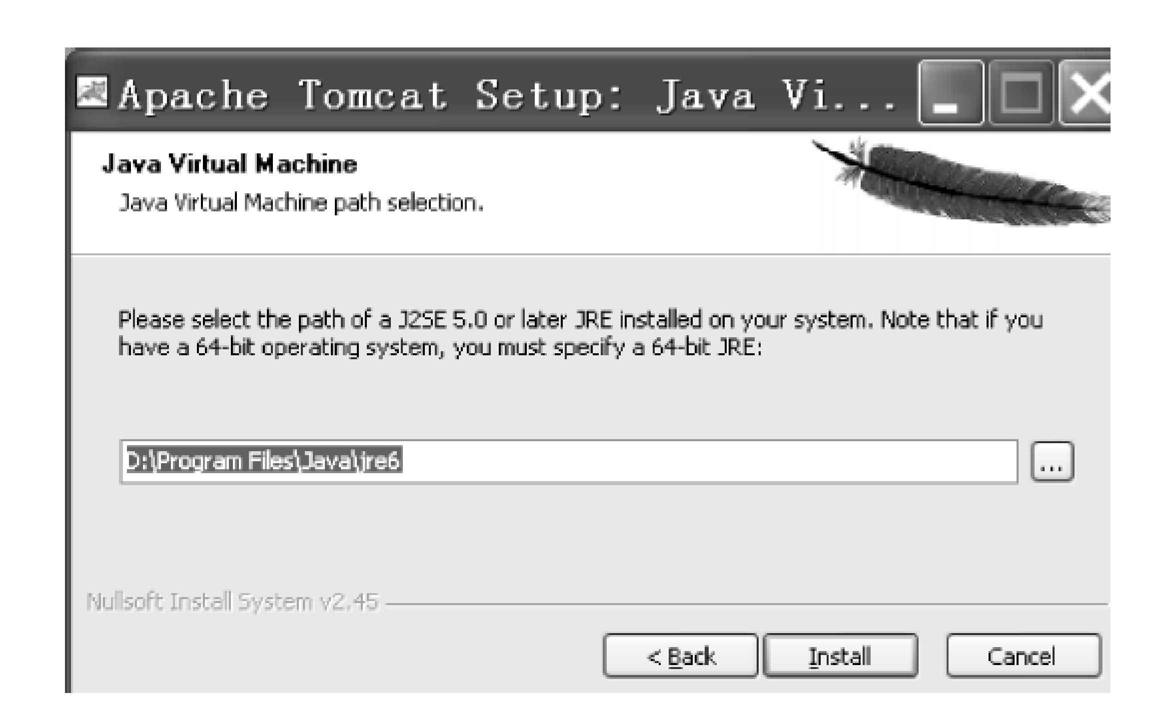


图 1-23 Tomcat 安装中 jre 的路径选择界面



（4）安装完成出现界面如图 1-24 所示。

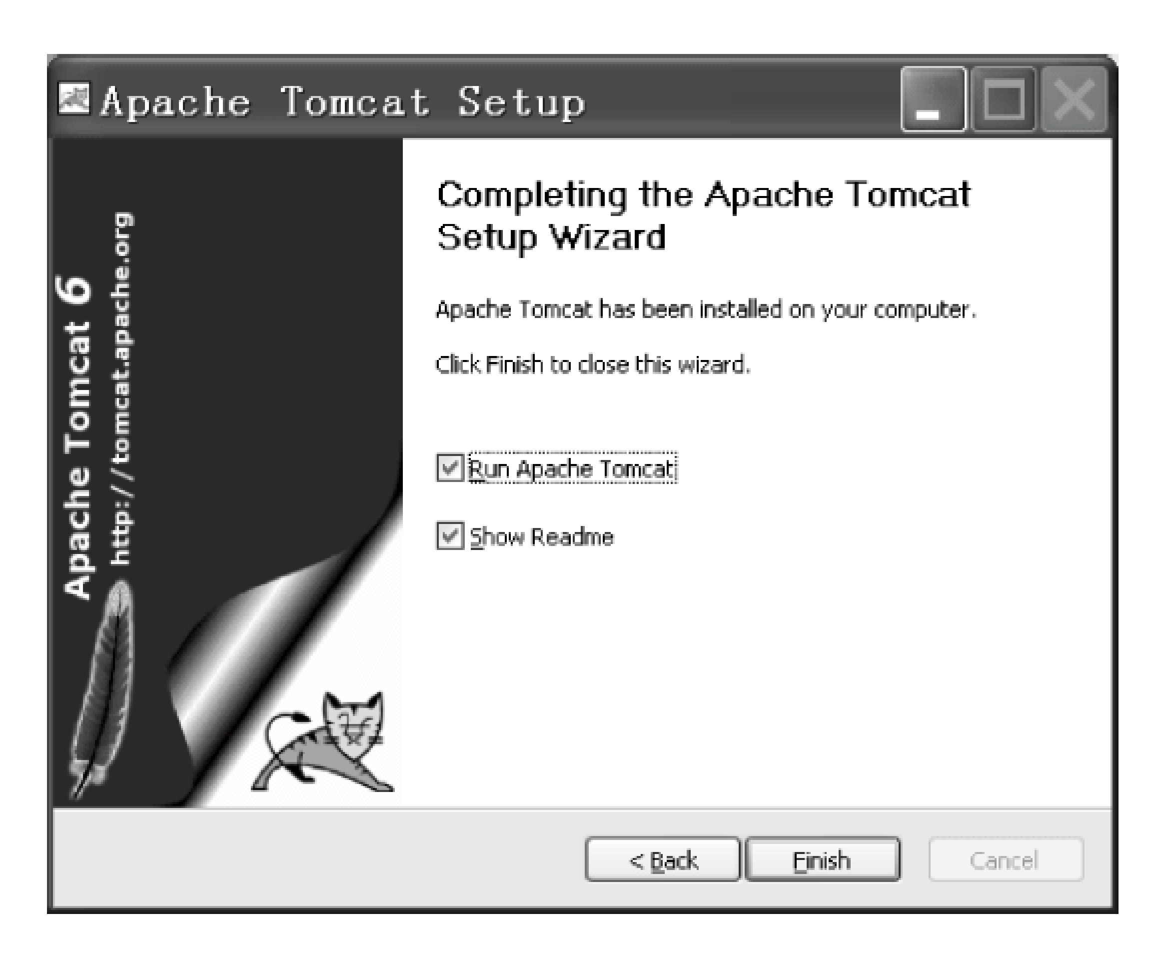


图 1-24 Tomcat 安装完成界面

3. Tomcat 的目录结构

/bin：存放 windows 或 Linux 平台上启动和关闭 Tomcat 的脚本文件。

/conf：存放 Tomcat 服务器的各种全局配置文件，其中最重要的是 server.xml 和 web.xml。

/doc：存放 Tomcat 文档。

/server：包含 classes、lib 和 webapps 三个子目录。

/server/lib：存放 Tomcat 服务器所需的各种 JAR 文件。

/server/webapps：存放 Tomcat 自带的两个 Web 应用 admin 应用和 manager 应用。

/common/lib：存放 Tomcat 服务器以及所有 Web 应用都可以访问的 jar 文件。

/shared/lib：存放所有 Web 应用都可以访问的 jar 文件(但是不能被 Tomcat 服务器访问)。

/logs：存放 Tomcat 执行时的日志文件。

/src：存放 Tomcat 的源代码。

/webapps：Tomcat 的主要 Web 发布目录，默认情况下把 Web 应用文件放于此目录。

/work：存放 JSP 编译后产生的 class 文件。

1.2.4 测试 Tomcat

1. 测试 Tomcat

Tomcat 启动后，在任务栏有图标，这是 Apache Tomcat 启动的标志。在浏览器的地址栏输入 http://127.0.0.1:8080 或 http://localhost:8080，会出现如图 1-25 所示的

欢迎界面,表示启动成功。

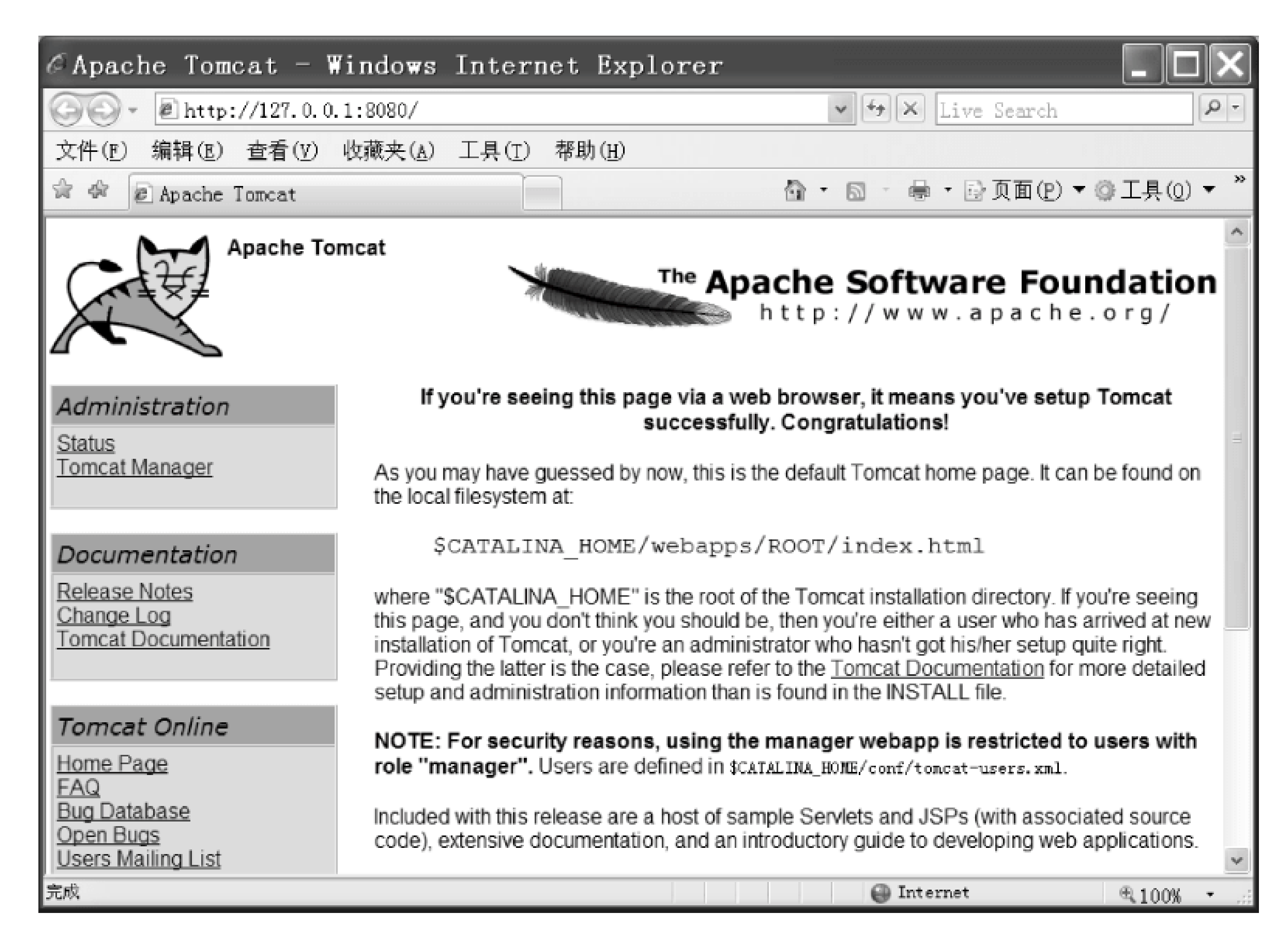


图 1-25 Apache Tomcat 欢迎界面

注意:127.0.0.1 是回送地址,指本地机,一般用来测试使用。在有类 IP 地址的规定中,第一部分是 1~126 为 A 类地址,128~191 为 B 类地址,那么中间留的 127.0.0.1 被称为本地回环地址,主要作用有两个:一个是测试本机的网络配置,能 PING 通 127.0.0.1 说明本机的网卡和 IP 协议安装都没有问题;另一个是某些 SERVER/CLIENT 的应用程序在运行时需调用服务器上的资源,一般要指定 SERVER 的 IP 地址,但当该程序要在同一台机器上运行而没有别的 SERVER 时就可以把 SERVER 的资源装在本机,SERVER 的 IP 地址设为 127.0.0.1 也同样可以运行。对于大多数习惯用 localhost 的用户来说,实质上就是指向 127.0.0.1 这个本地 IP 地址,在操作系统中有个配置文件将 localhost 与 127.0.0.1 绑定在了一起。

2. 增加虚拟服务端口号

如果安装好的 Tomcat 在端口上与其他开发工具有冲突,或者想重新找一个端口号,那么可以通过增加虚拟服务端口号的方法来解决。首先打开 D:\Program Files\Apache Software Foundation\Tomcat 6.0\conf 路径下的 server.xml,可以使用任何一种文本编辑器打开,找到下面这段语句:

```
<Connector port="8080" protocol="HTTP/1.1"connectionTimeout="20000" redirectPort="8443" />
```

然后把这一段复制并粘贴在原语句的后面,把复制的这段语句中 8080 改为 2010 或其他不冲突的端口号,最后保存并关闭 server.xml 文件,重启 Tomcat。在浏览器的地址栏输入 http://127.0.0.1:2010,如果出现如图 1-25 所示的页面,即为修改成功。

1.3　JSP 开发工具 MyEclipse

MyEclipse 企业级工作平台(MyEclipse Enterprise Workbench ，MyEclipse)是对 Eclipse IDE 的扩展，利用它用户可以在数据库和 Java EE 的开发、发布，以及应用程序服务器的整合方面极大地提高工作效率。MyEclipse 是功能丰富的 Java EE 集成开发环境，包括了完备的编码、调试、测试和发布功能，完整支持 HTML，Struts，JSF，CSS，Javascript，SQL，Hibernate。

在结构上，MyEclipse 的特征可以被分为以下 7 类：

(1) Java EE 模型；

(2) Web 开发工具；

(3) EJB 开发工具；

(4) 应用程序服务器的连接器；

(5) Java EE 项目部署服务；

(6) 数据库服务；

(7) MyEclipse 整合帮助。

简单而言，MyEclipse 是 Eclipse 的插件，也是一款功能强大的 Java EE 集成开发环境，支持代码编写、配置、测试以及除错，MyEclipse 6.0 以前版本需先安装 Eclipse。MyEclipse 6.0 以后版本安装时不需安装 Eclipse。

1.3.1　MyEclipse 的下载与安装

(1) 下载 MyEclipse 版本配合 JDK 1.6 和 Tomcat 6.0 使用。

(2) 双击运行下载的文件 MyEclipse_6.0.1GA_E3.3.1_Installer.exe，如图 1-26 所示，准备工作好了以后，出现如图 1-27 所示的安装提示界面。

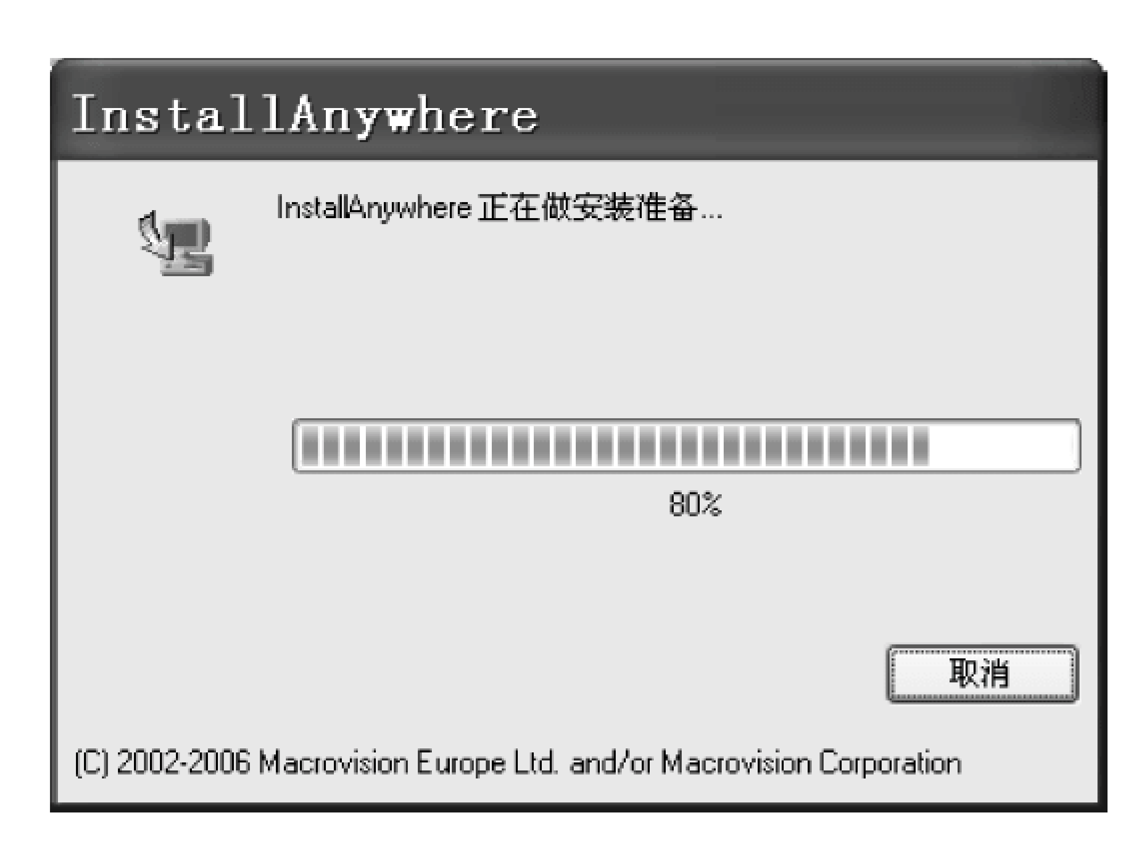


图 1-26　MyEclipse 安装准备界面

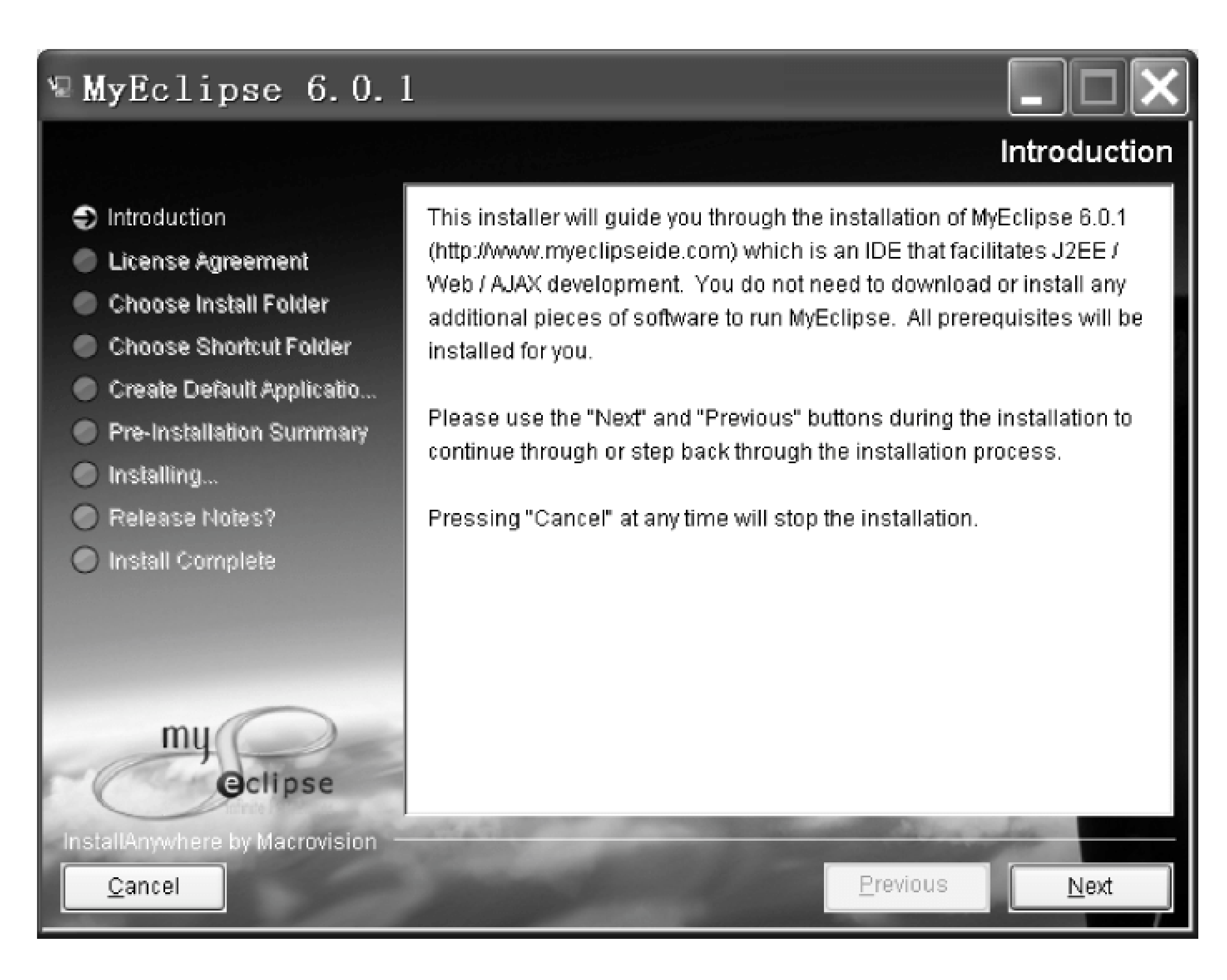


图 1-27　MyEclipse 安装界面

(3) 先单击 Next,接受协议,再单击 Next,如图 1-28 所示,修改好 MyEclipse 的目录之后,再单击 Next,选择 MyEclipse 的安装目录之后,继续单击 Next,直到最后完成 MyEclipse 的安装。

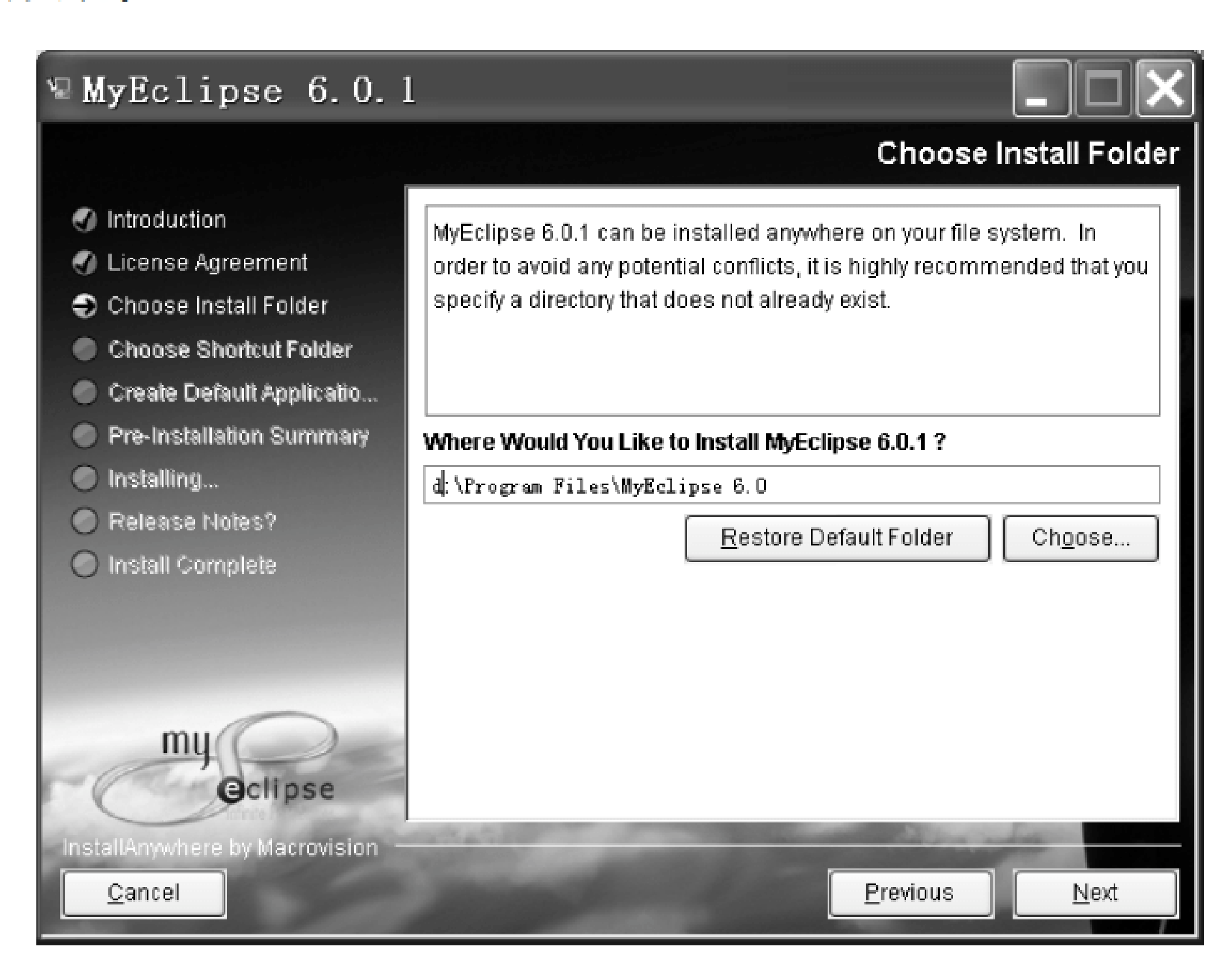


图 1-28　MyEclipse 安装路径选择界面

(4) 启动 MyEclipse,首先要选择 MyEclipse 的工作空间,工作空间存放项目文件和一些配置信息,这里选择 D:\Program Files\Apache Software Foundation\Tomcat 6.0\webapps 作为工作空间,如图 1-29 所示。启动以后页面如图 1-30 所示。

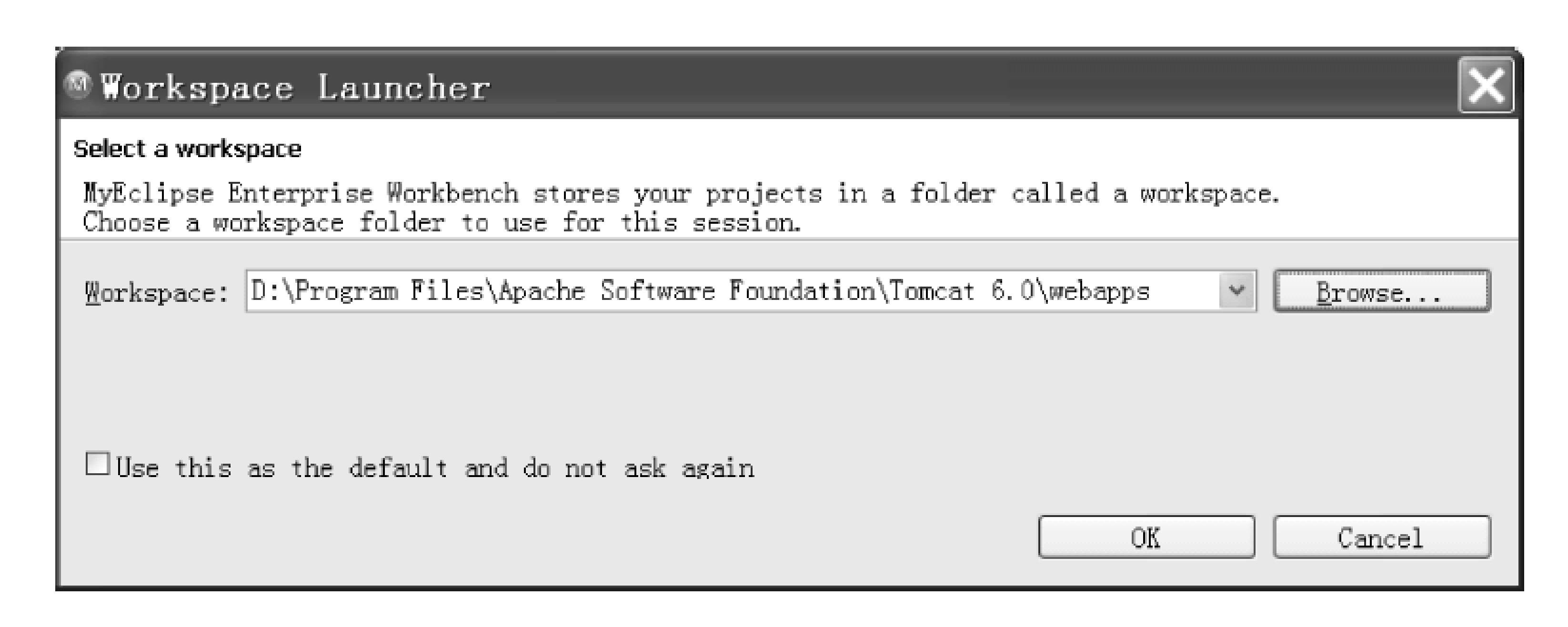


图 1-29　MyEclipse 的工作空间设置界面

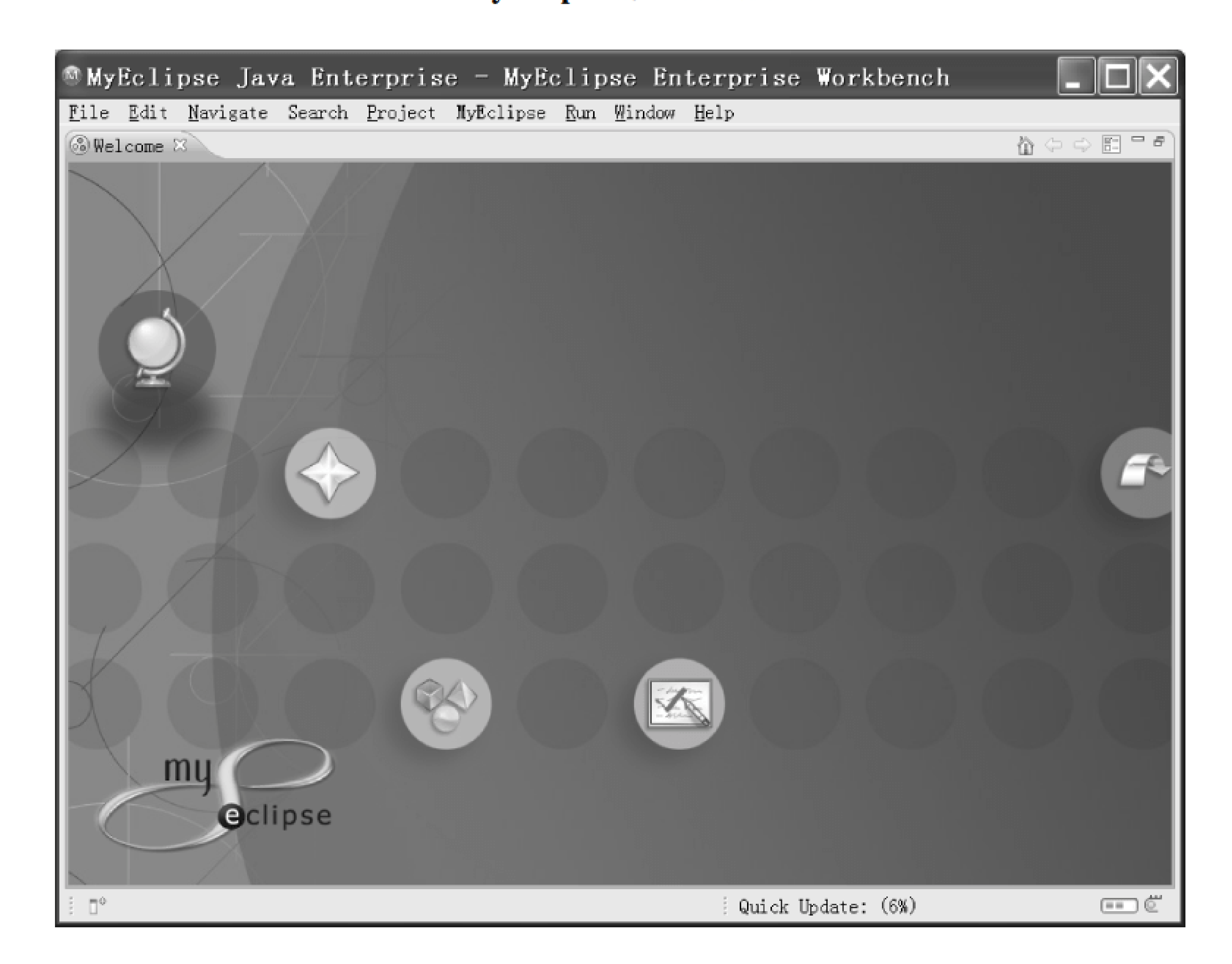


图 1-30　MyEclipse 工作界面

1.3.2　在 MyEclipse 中配置 Tomcat 服务器

（1）启动 MyEclipse 之后选择 MyEclipse 菜单的 Preferences 选项，如图 1-31 所示。弹出 Preferences 对话框如图 1-32 所示，打开 MyEclipse 左侧的加号。

（2）打开 MyEclipse 下面的 Servers 左侧的加号，找到 Tomcat，选择你所安装的 Tomcat 对应的版本，如图 1-33 所示，选择了 Tomcat 5.x，对话框右侧出现对应的可设置项目，单击 Tomcat home directory 右边的 Browse 按钮，弹出对话框如图 1-34 所示，选择本机 Tomcat 的安装路径，选到 Tomcat 文件夹为止，然后单击“确定”按钮。

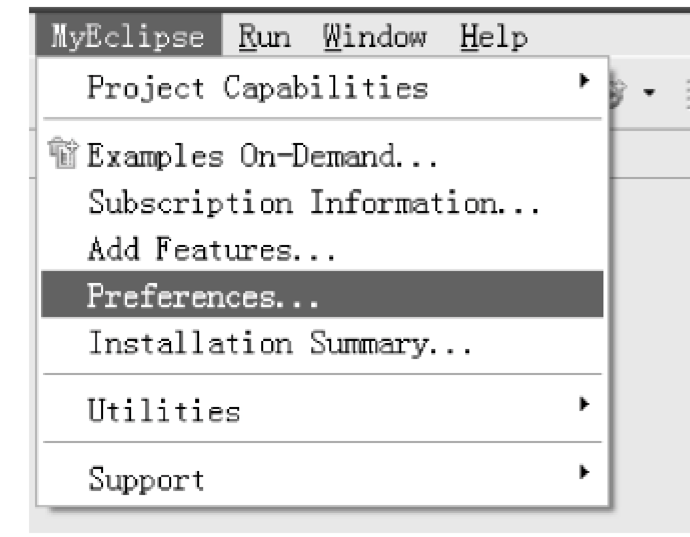


图 1-31　选择 MyEclipse 菜单的 Preferences 选项

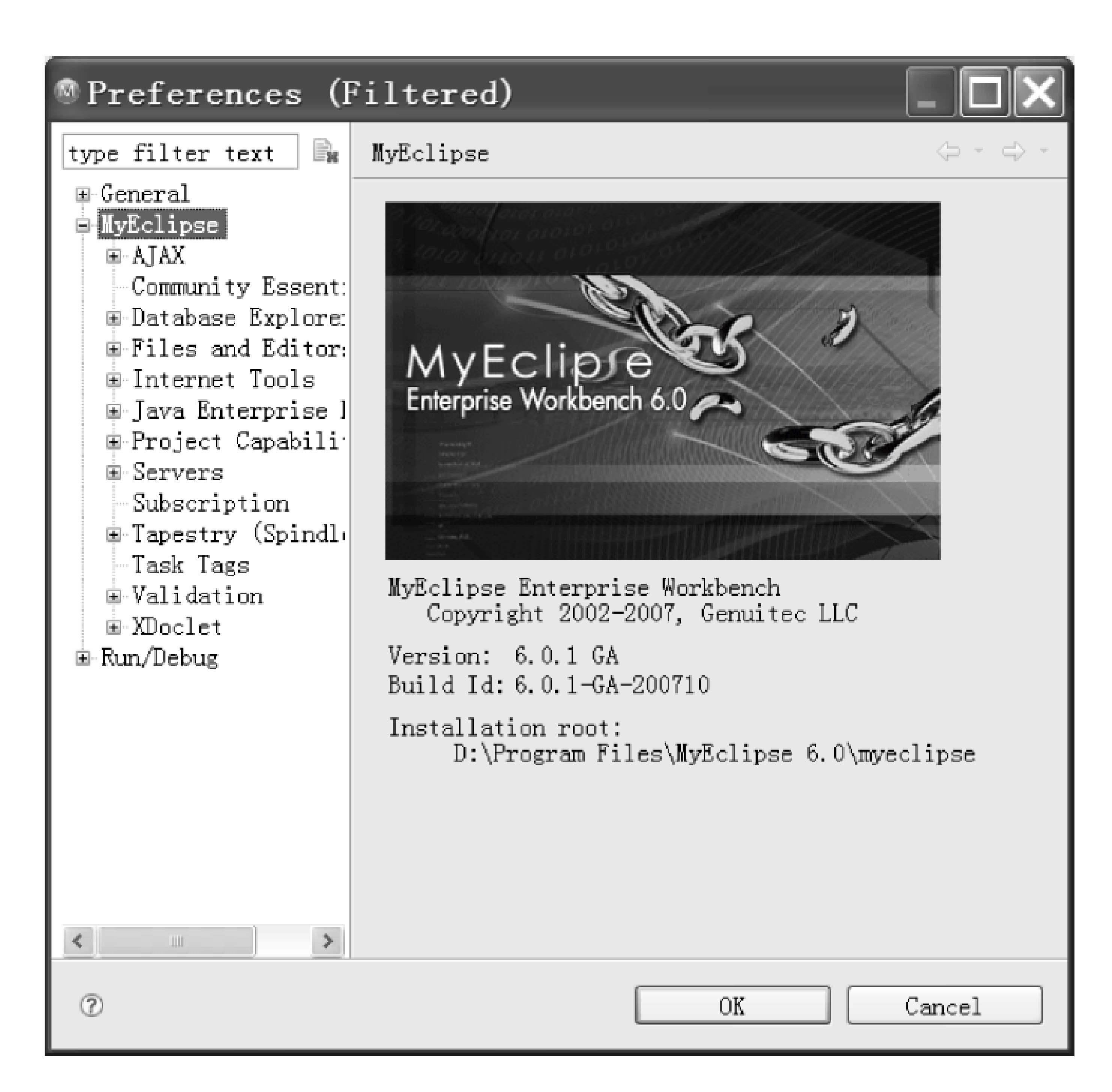


图 1-32 Preferences 对话框

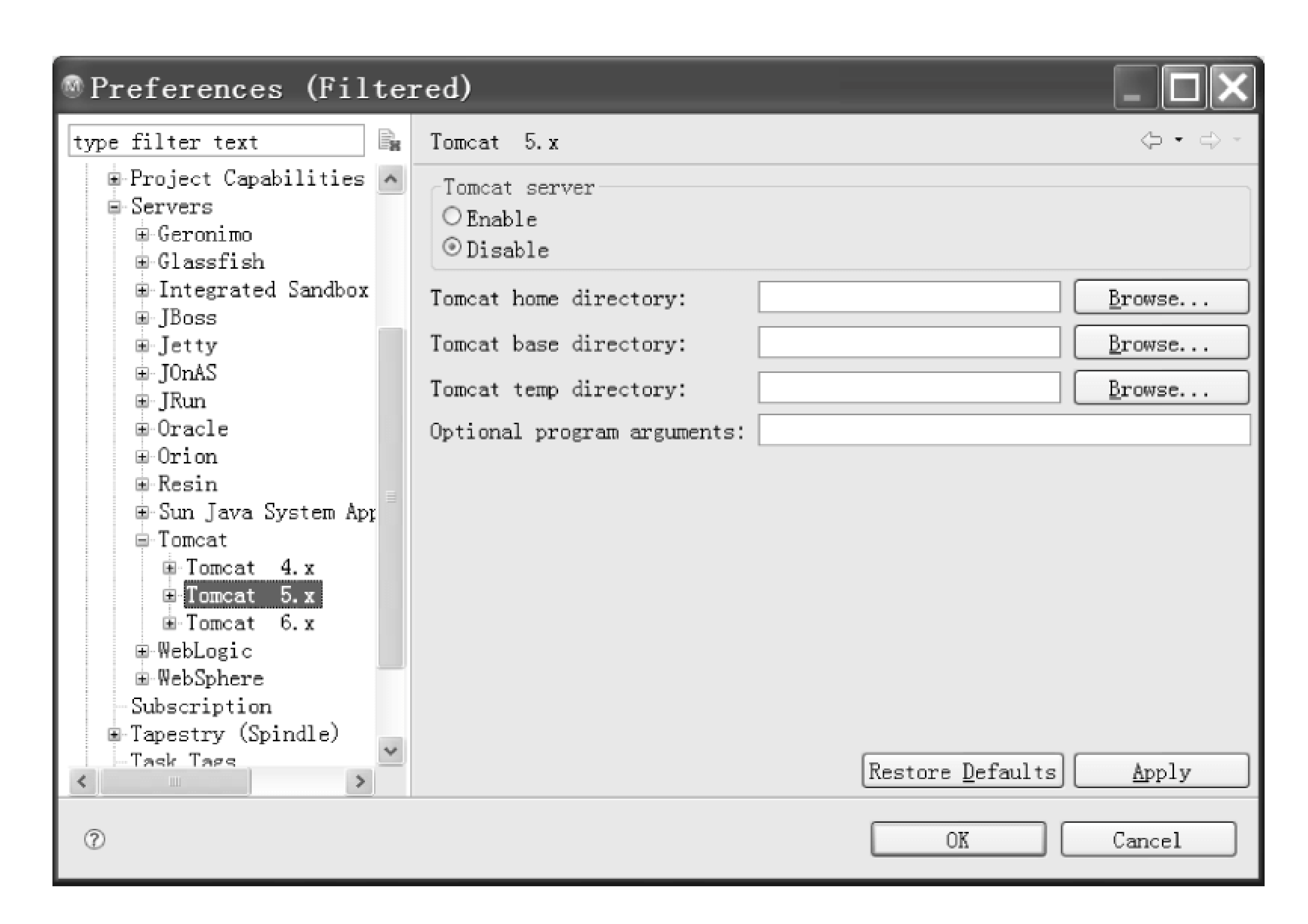


图 1-33 选择本机安装的 Tomcat 版本

(3) 设置好 Tomcat home directory 之后，下面两项 Tomcat base directory 和 Tomcat temp directory 自动识别，然后把上面的单选按钮改为 Enable，如图 1-35 所示。

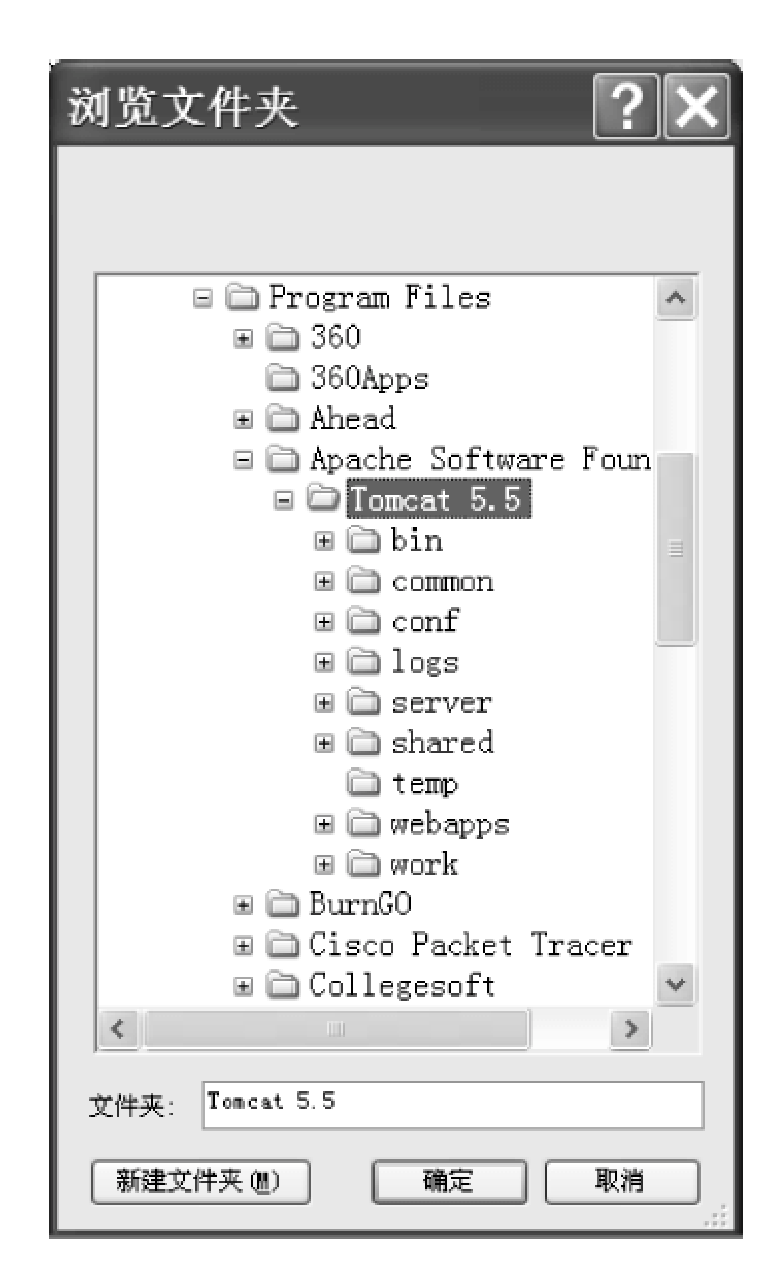


图 1-34　选择 Tomcat 安装的路径

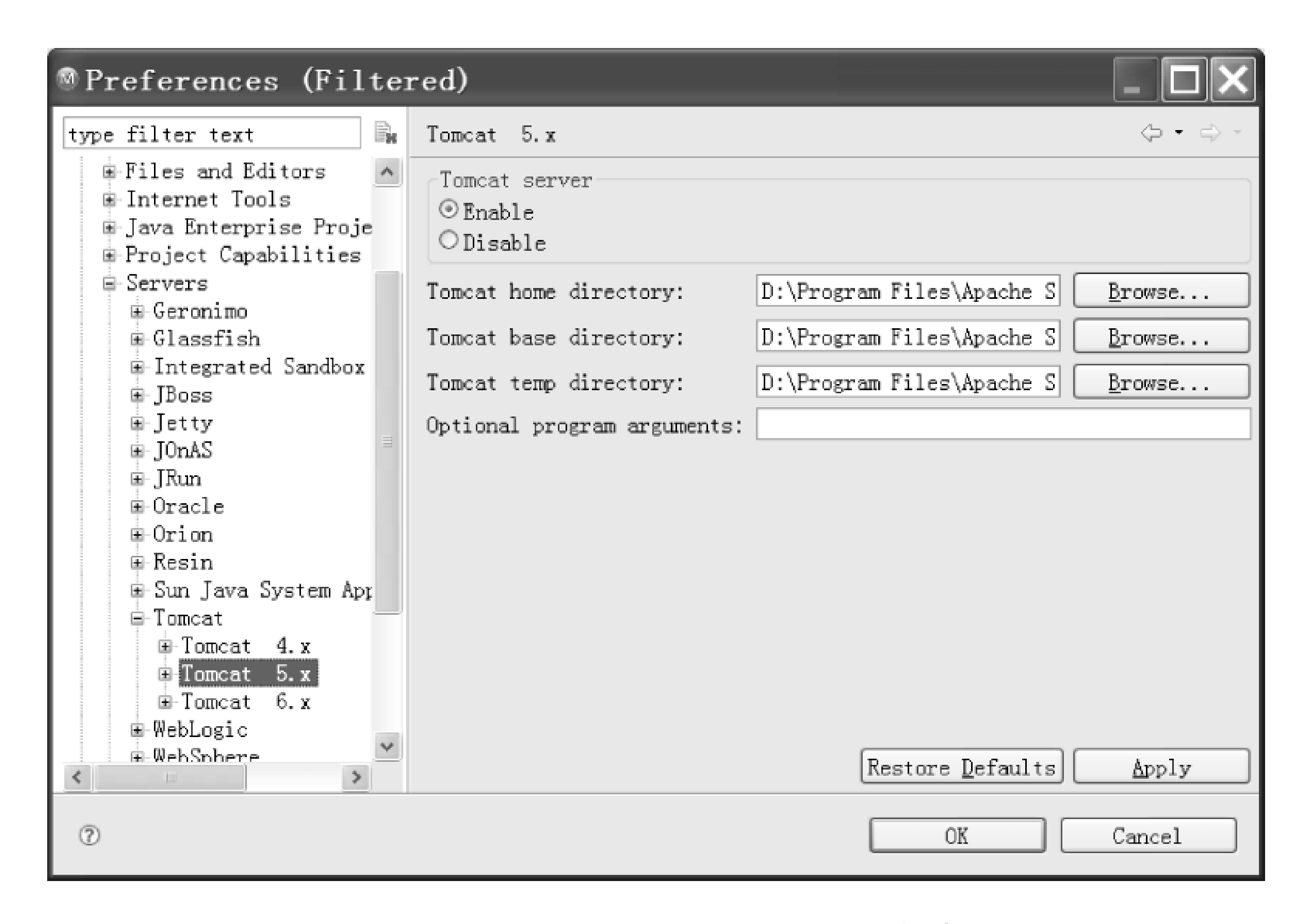


图 1-35　设置 Tomcat server 为可用状态

(4) 以上 Tomcat 设置好了，下面打开 Tomcat 左侧加号，如图 1-36 所示，开始设置 JDK，单击 Add 按钮，弹出对话框 Tomcat＞Add JVM，如图 1-37 所示，在这里要指定 JRE home directory，单击 Browse 按钮，在“浏览文件夹”对话框中选择，Java 安装路径下的 jdk 文件夹，如图 1-38 所示。

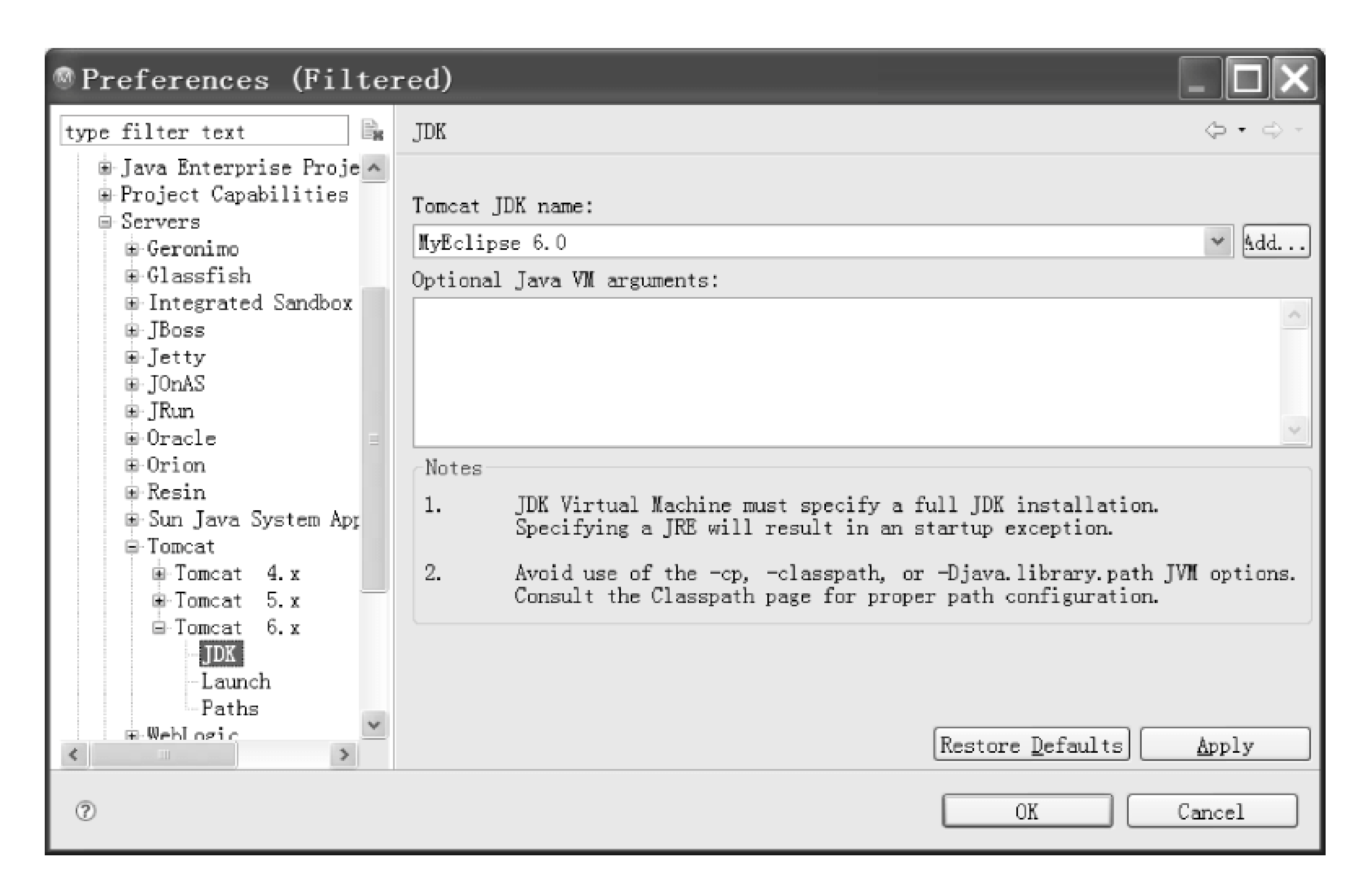


图 1-36 设置 Tomcat 对应的 JDK

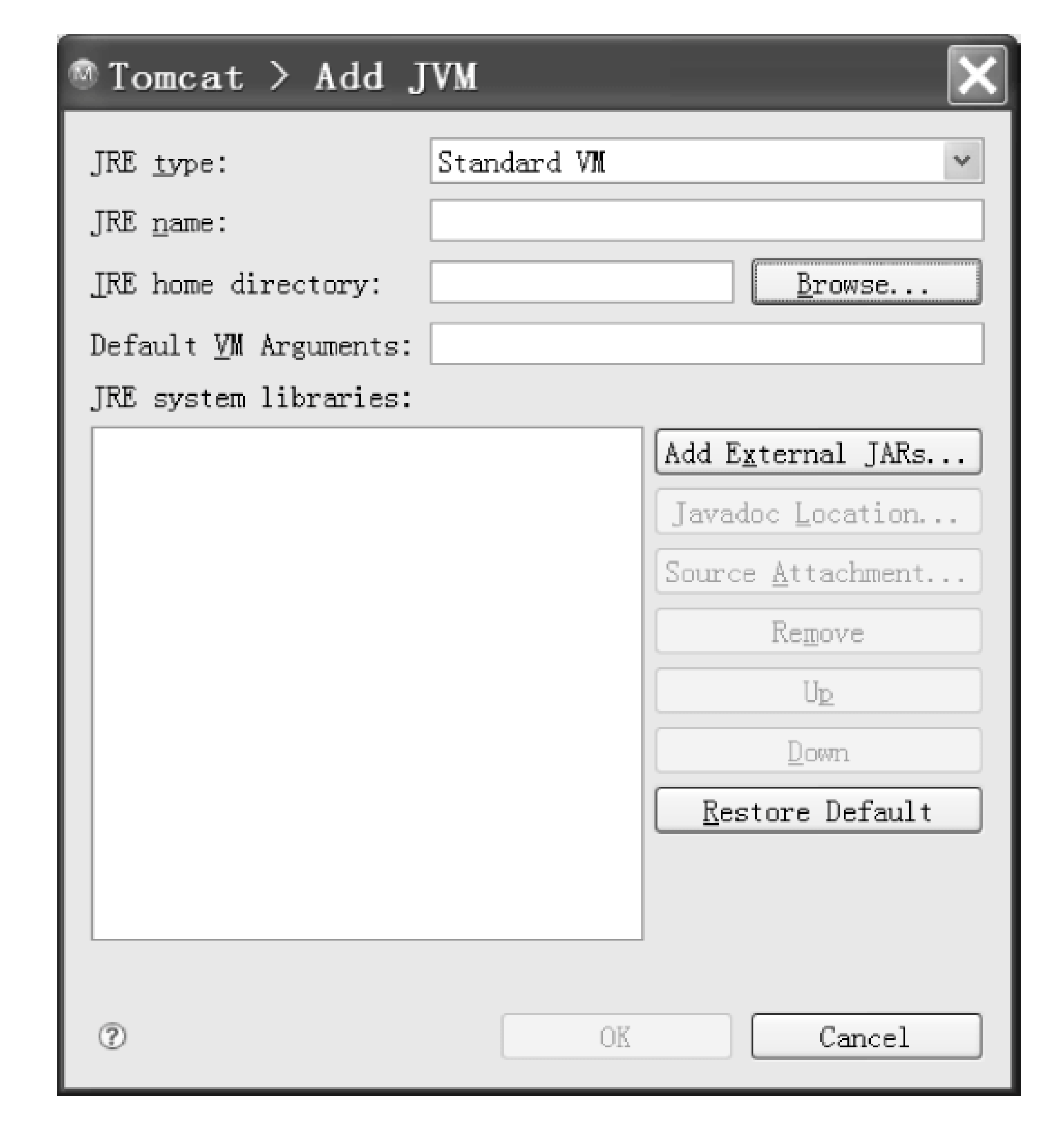


图 1-37 添加 JVM

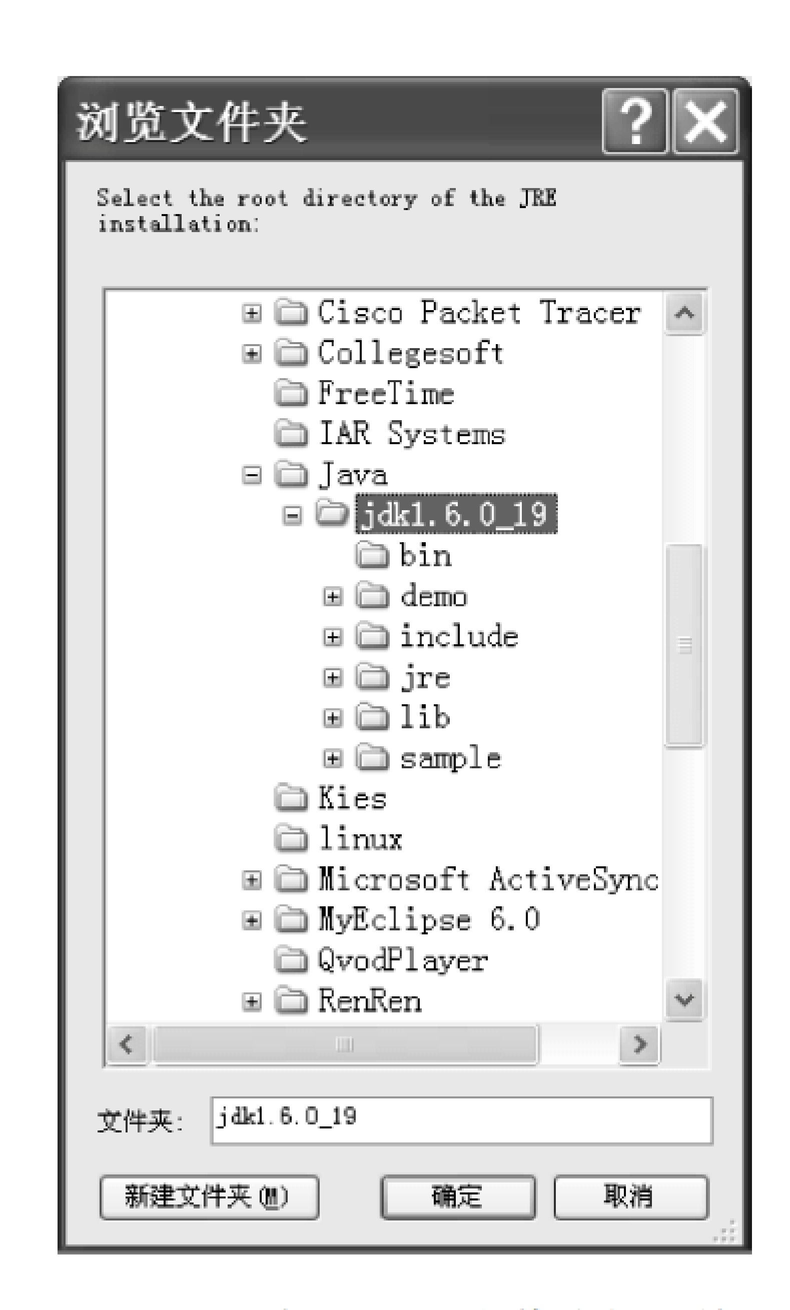


图 1-38 找到 Java 安装路径下的 jdk 文件夹

(5) 选择好路径之后 JRE name 和 JRE system libraries 自动识别,如图 1-39 所示,单击 OK 按钮,至此,JDK 设置完毕,如图 1-40 所示,单击 OK 按钮。

(6) 所有关于 MyEclipse 中服务器的配置完成,图 1-41 中 Tomcat 5.x 就是刚刚添加好的服务器,现在它还处于停止状态,运行服务器,如图 1-42 所示服务器正常启动。

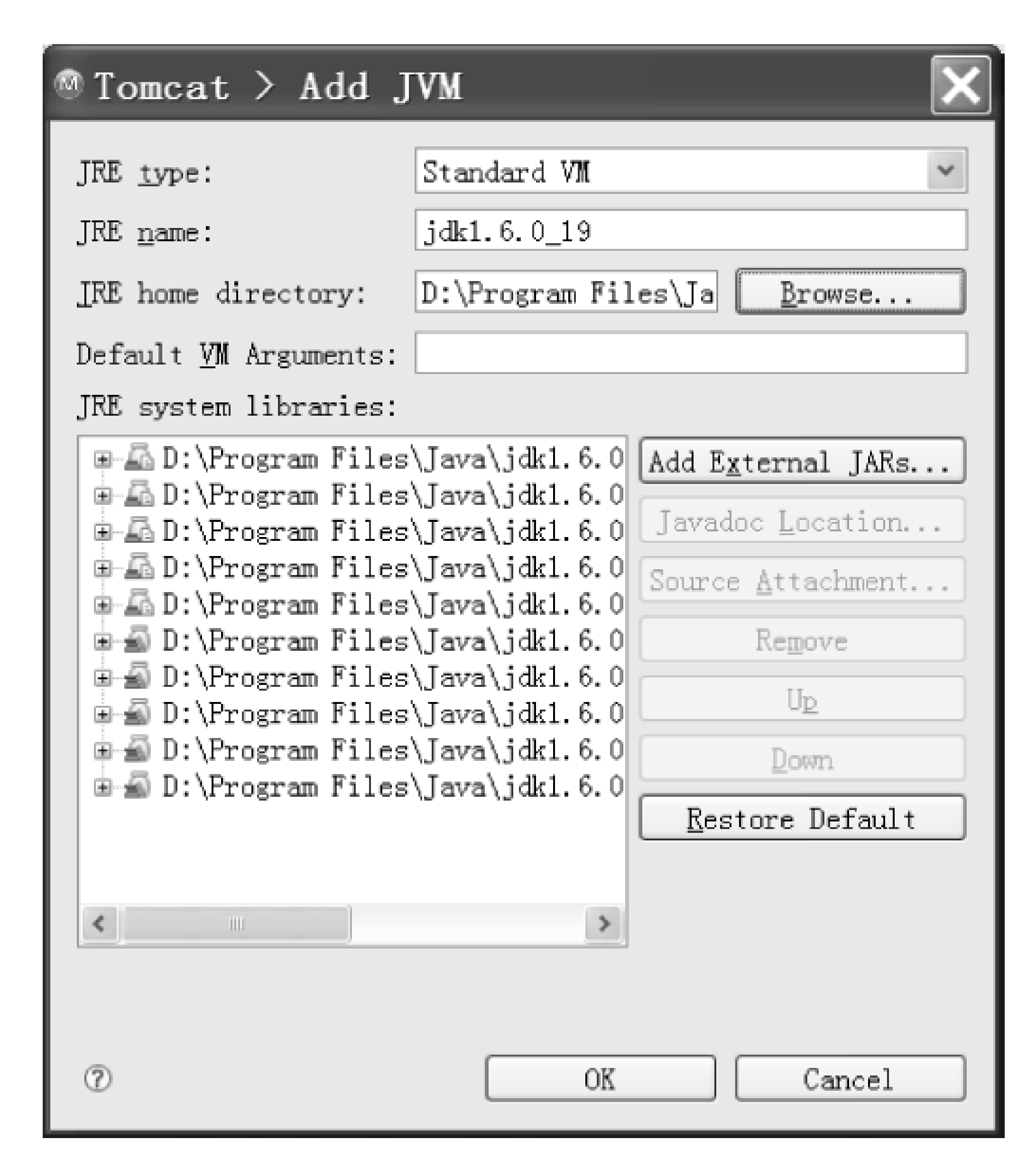


图 1-39 JRE name 和 JRE system libraries 自动识别

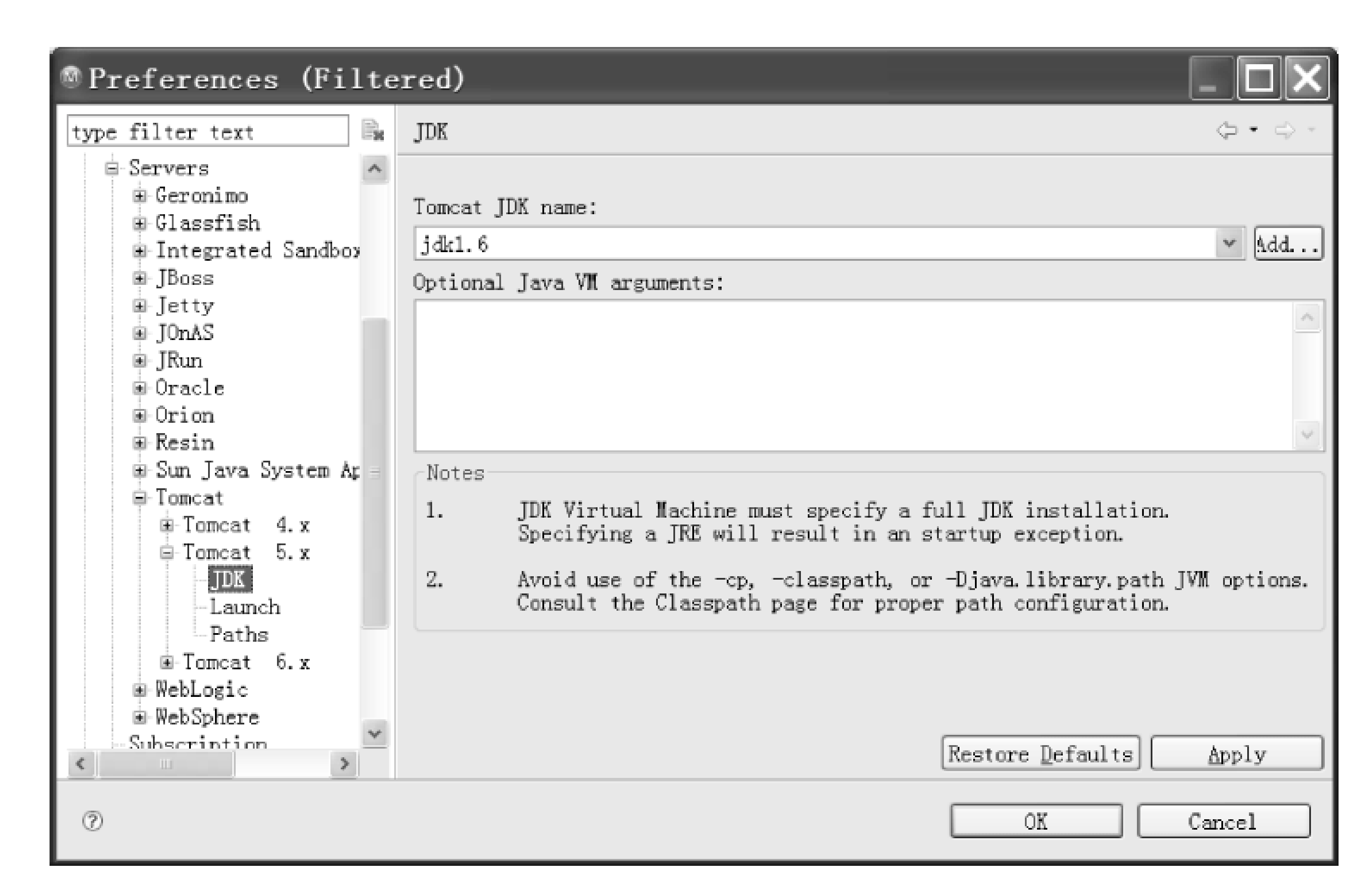


图 1-40 JDK 设置完毕

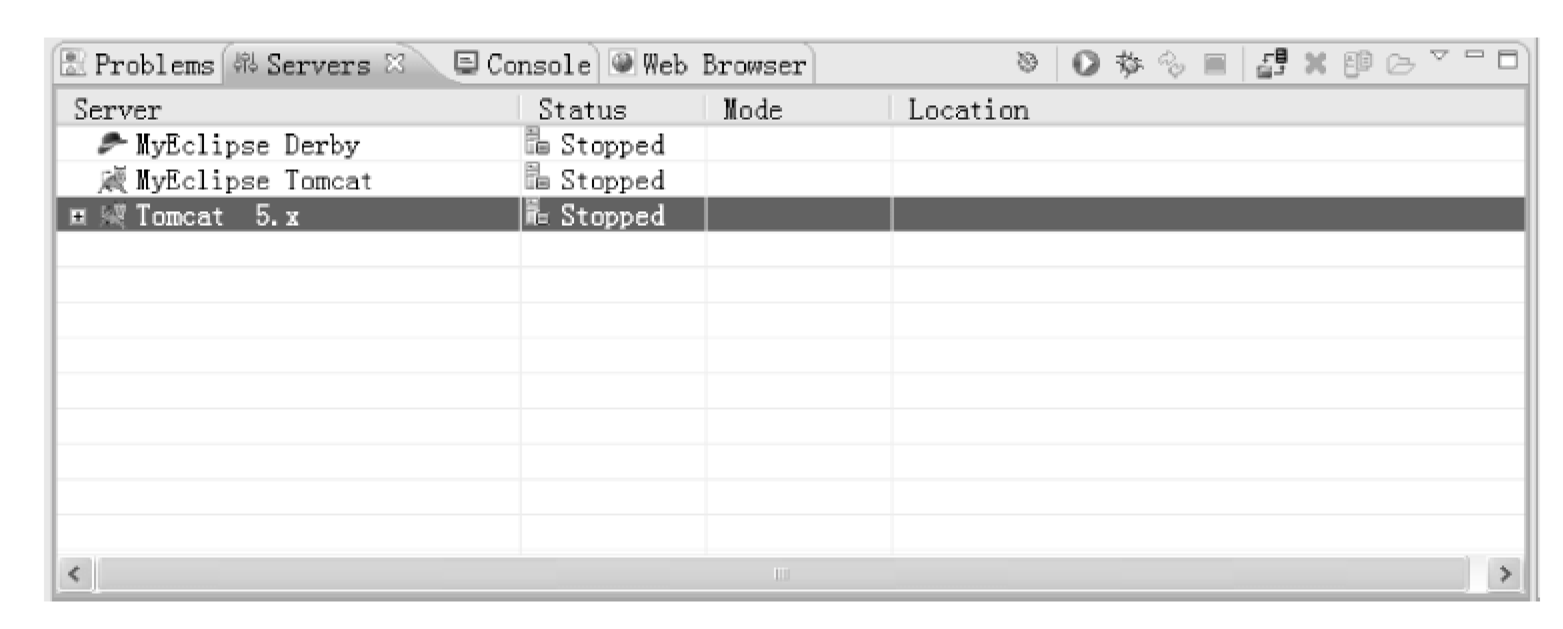


图 1-41 成功添加 Tomcat 5. x

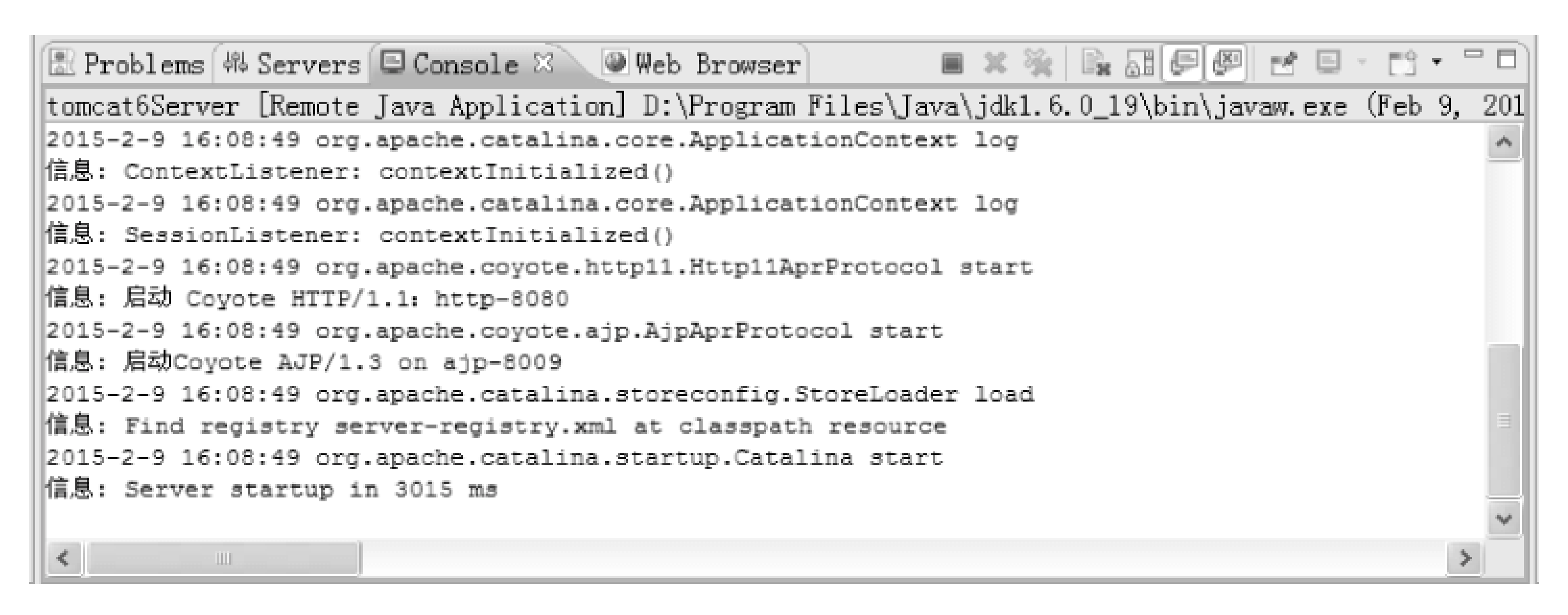


图 1-42　在 MyEclipse 中 Tomcat 5.x 正常运行

1.3.3　使用 MyEclipse 创建第一个 JSP 程序

（1）启动 MyEclipse，首先调整工作空间为 Tomcat 的 ROOT 目录下，如图 1-43 所示，以后的工作空间都以此为准。

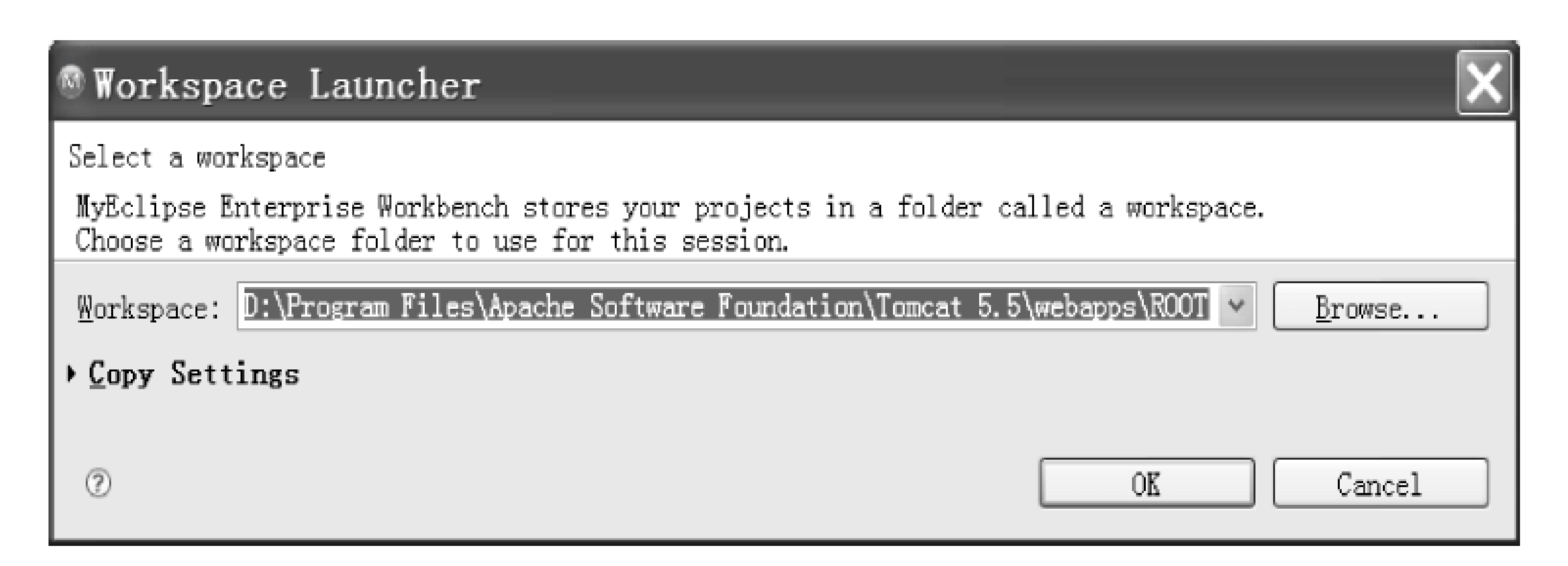


图 1-43　设置工作空间为 Tomcat 的 ROOT 目录下

（2）启动 MyEclipse 工具，新建一个工程，File-New-Web Project，弹出 New Web Project 窗口，如图 1-44 所示，Project Name 设置为 h，然后单击 Finish 按钮完成工程的新建。

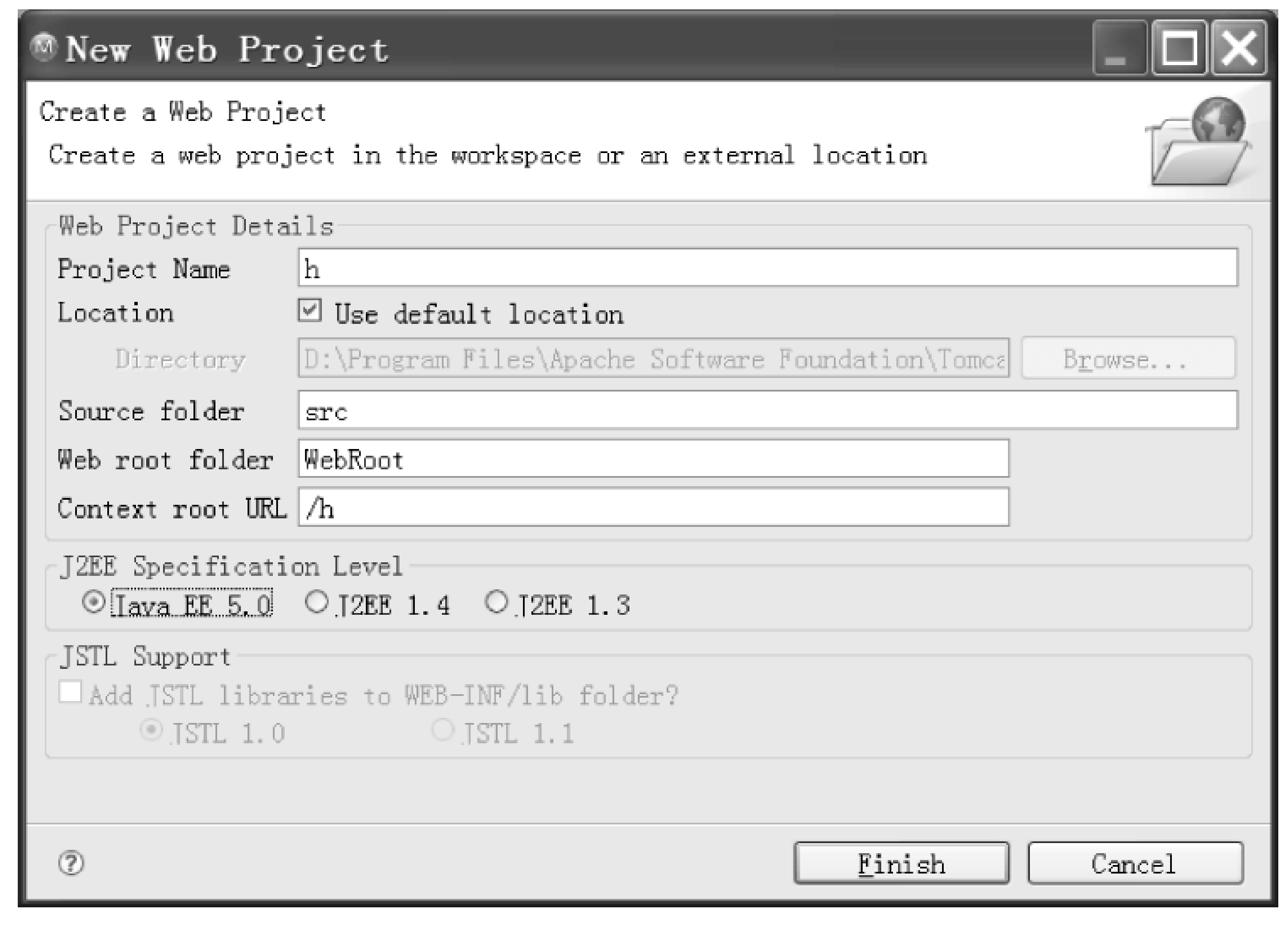


图 1-44　新建一个工程 h

（3）在服务器中添加刚刚建立好的工程 h，单击图 1-45 中的按钮，就会弹出 Server Deployments 对话框，如图 1-46 所示，单击右侧的 Add 按钮。

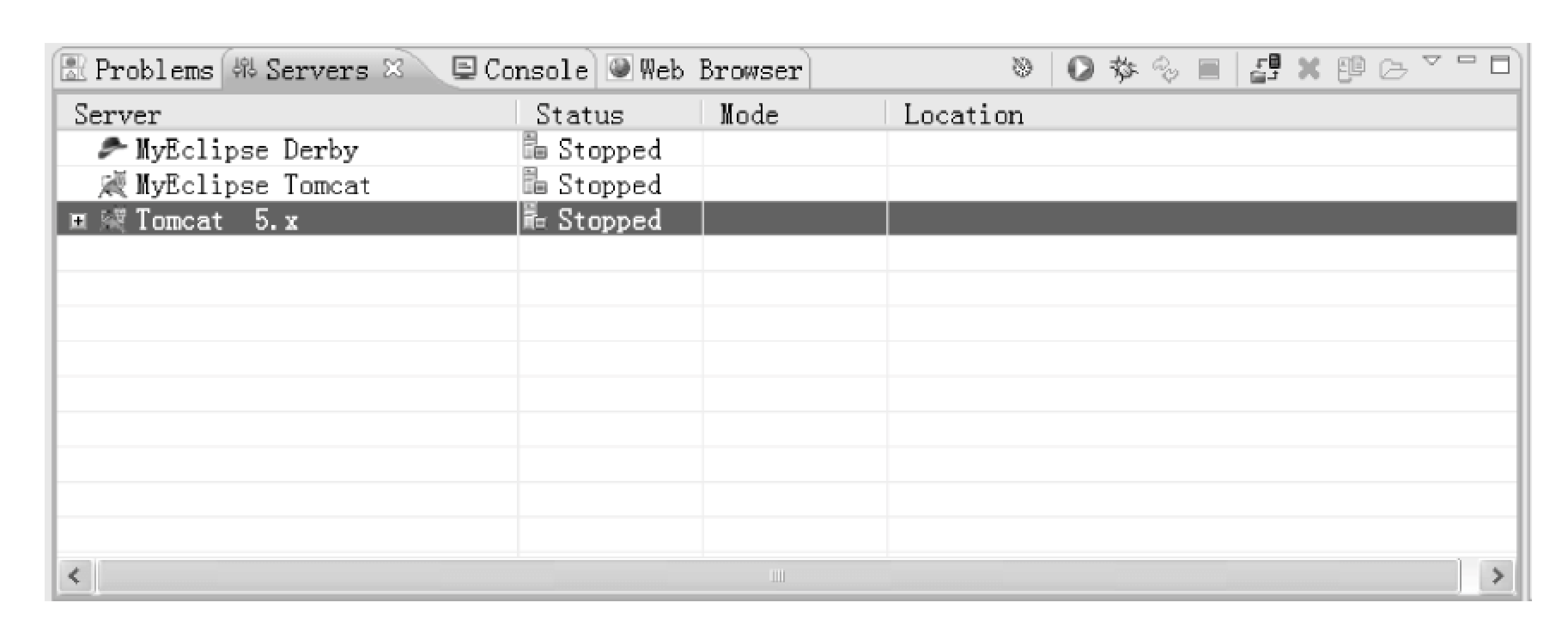


图 1-45　向服务器中添加新建的工程

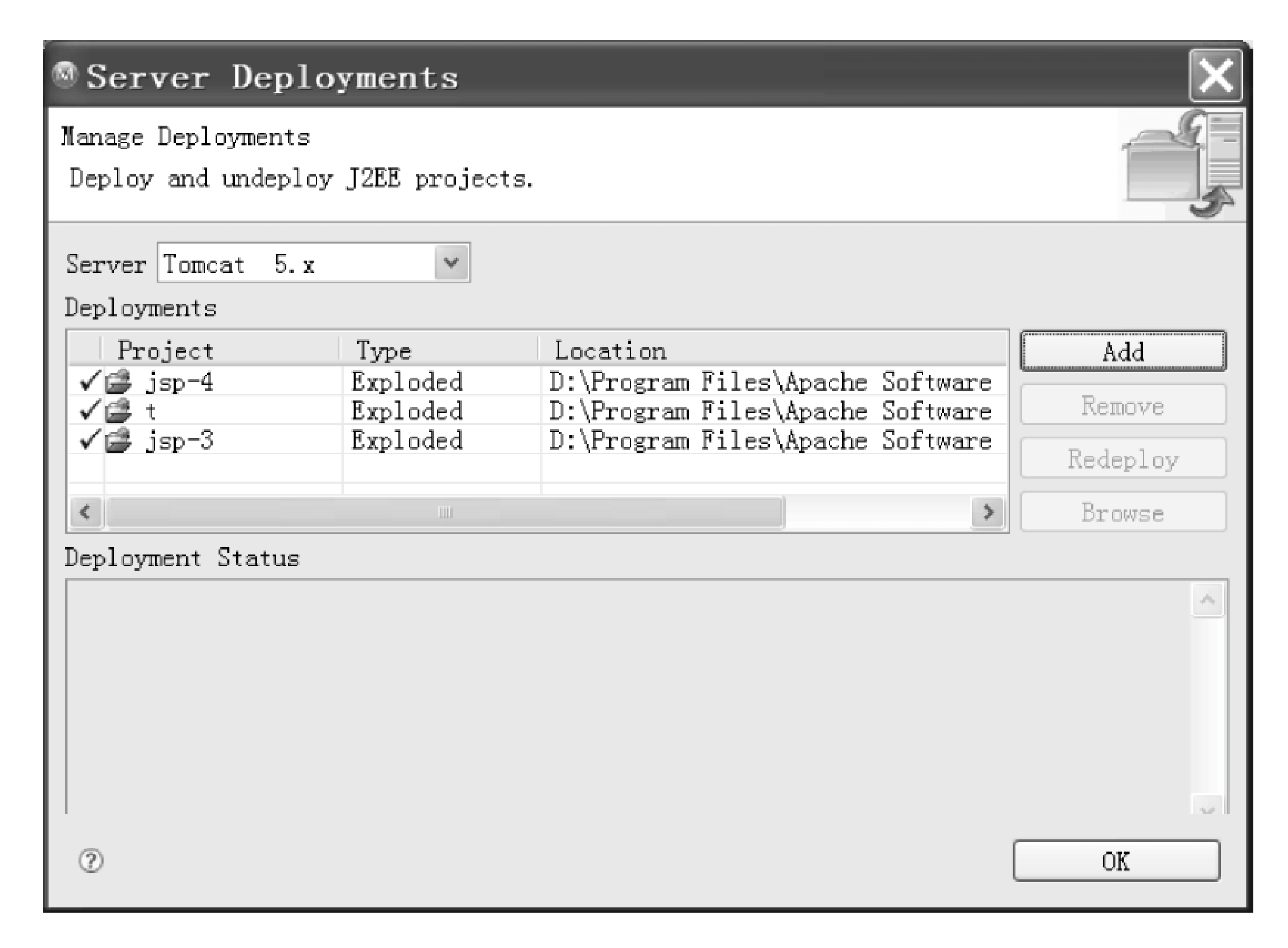


图 1-46　Server Deployments 对话框

（4）弹出 New Deployment 对话框，选择需要添加的工程，如图 1-47 所示，然后单击 Finish 按钮就会看到选择的工程 h 已经添加到了 Deployments 列表中，如图 1-48 所示，单击 OK 按钮。

（5）现在在 Server 标签中已经可以看到工程 h 已经在运行的服务器下了，如图 1-49 所示，然后转换到 Web Browser 标签，在地址栏输入地址 http://127.0.0.1:8080/h 就可以看到我们的第一个 JSP 工程的运行了，如图 1-50 所示，这个是项目创建时自动生成的默认首页。

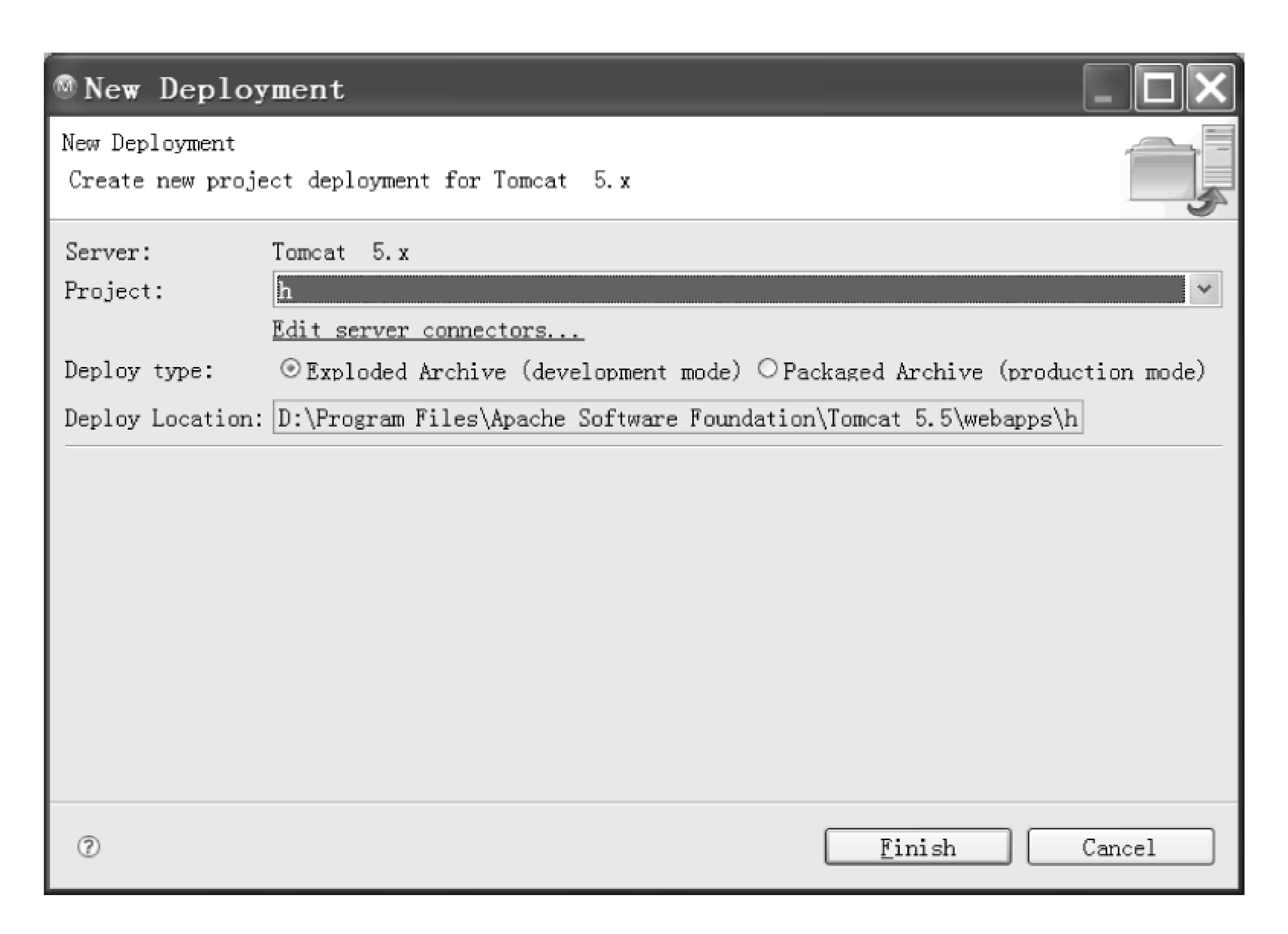


图 1-47 New Deployment 对话框选择 h 工程

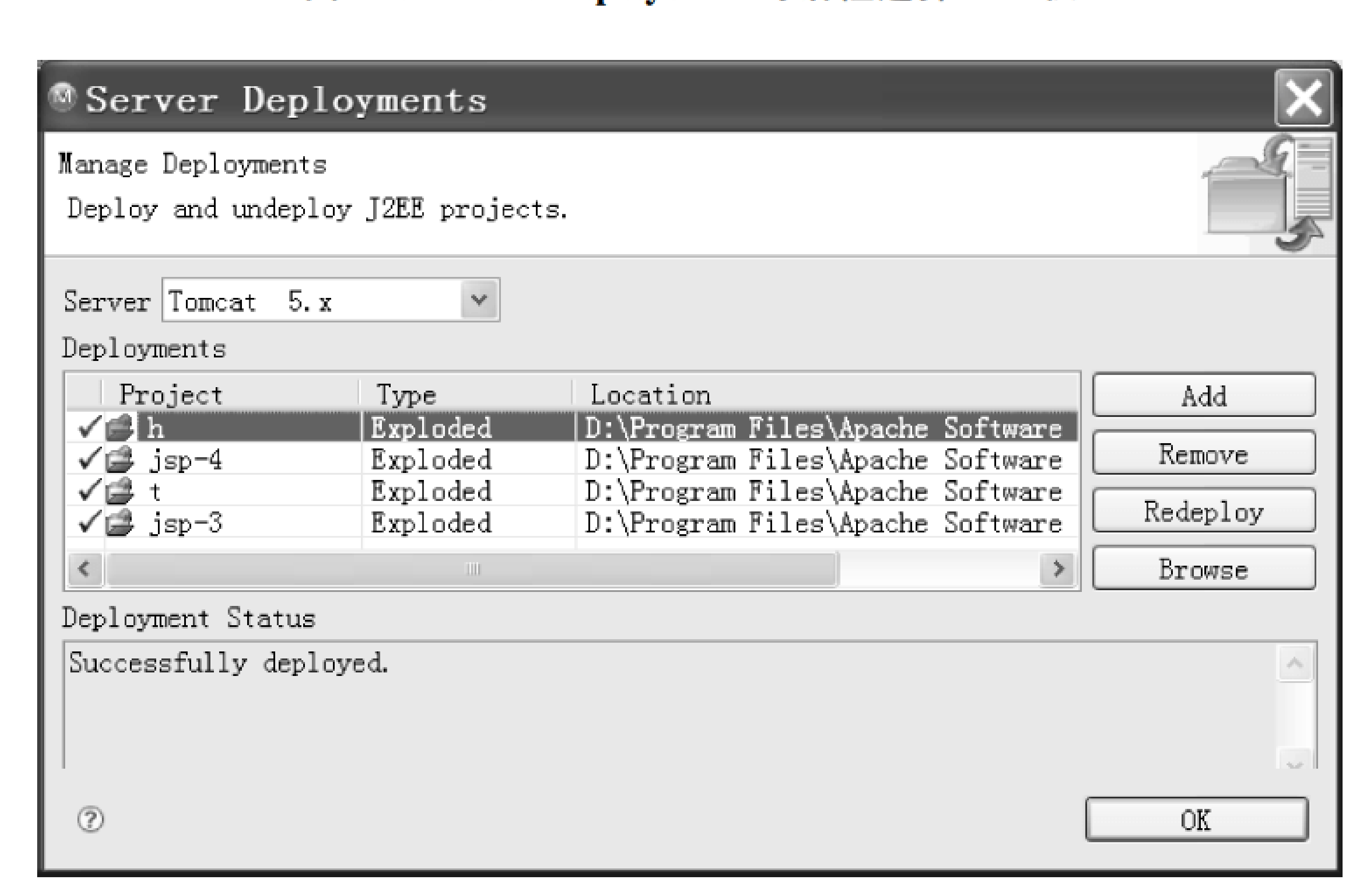


图 1-48 工程 h 已经添加到了 Deployments 列表

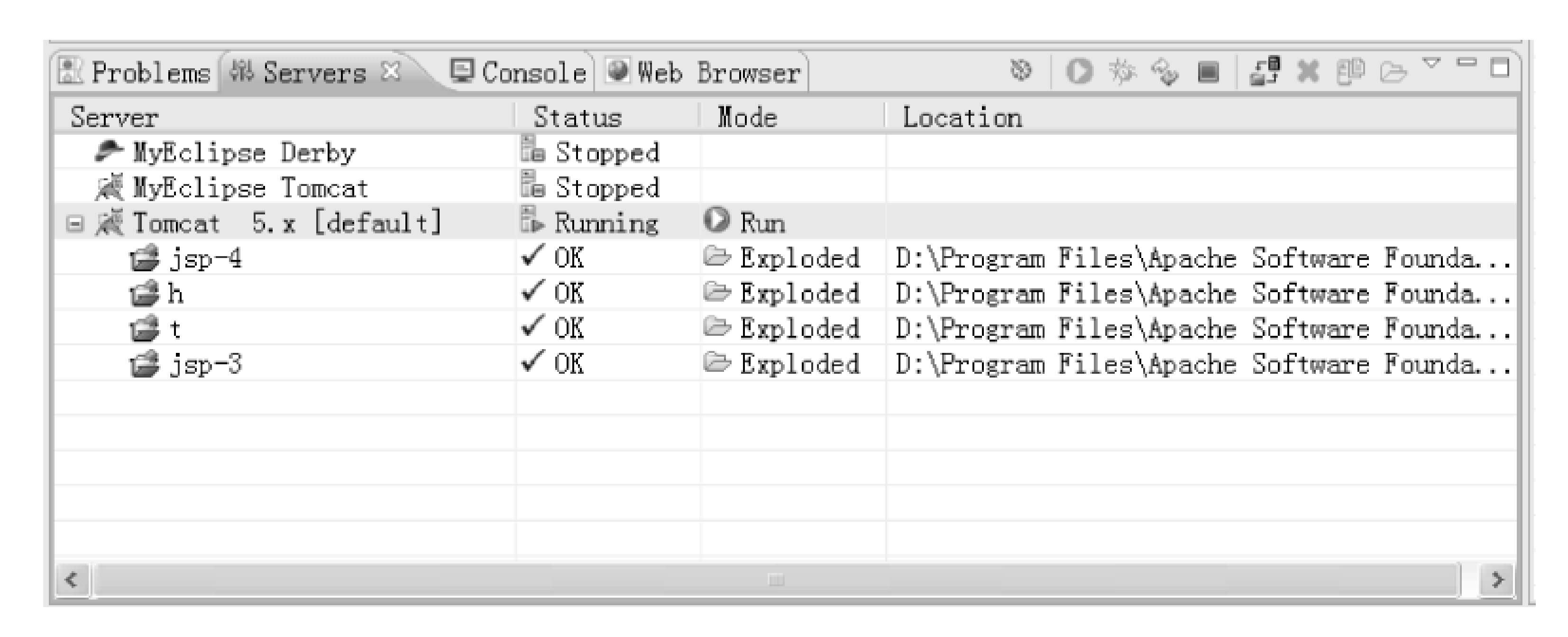


图 1-49 工程 h 已经在服务器中运行

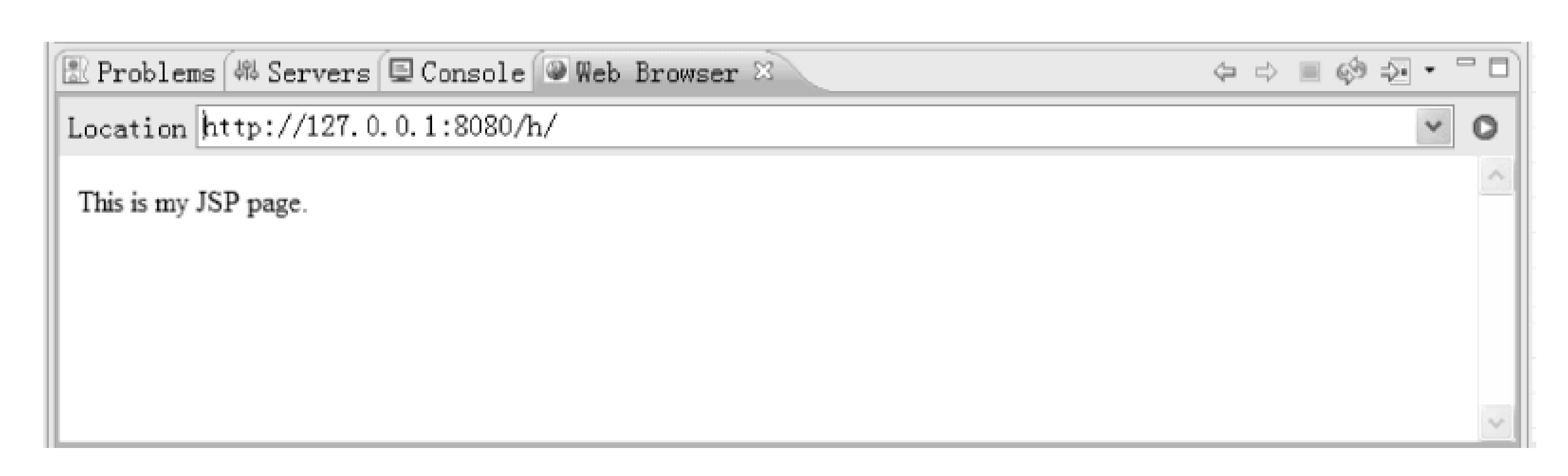


图 1-50　在浏览器中测试工程

1.4　设置 Web 服务目录

1.4.1　Web 服务目录

Web 服务目录就是用户浏览器能够访问的页面所在的目录。如果要发布网页，必须将编写好的 JSP 页面放到 Web 服务目录中。在本章中安装的 Tomcat 的根目录在服务器上的路径为 D:\Program Files\Apache Software Foundation\Tomcat 6.0\webapps\ROOT。webapps 子目录下除了 ROOT 以外，还有 docs、examples、manager 等子目录，这些子目录也是可以被访问的，比如，在浏览器的地址栏输入 http://127.0.0.1:8080/docs/apr.html，就可以访问 docs 子目录中的 apr.html 文件。除了这些默认安装的服务子目录以外，用户还可以在 webapps 目录中创建新的子目录，例如，上一节中建立的 h 项目就是与 webapps 目录并列的子目录。

1.4.2　增加虚拟服务目录

除了在安装目录中的 webapps 目录下创建 Web 服务子目录外，如果想在一个方便的目录下存放 Web 项目，那么就可以通过增加虚拟服务目录的方法来解决。首先打开 D:\Program Files\Apache Software Foundation\Tomcat 6.0\conf 路径下的 server.xml，在文件倒数第四行找到</host>，在其前面插入语句：

```
<Context path="/test" docBase="d:\test\" debug="0" reloadable="true">
</Context>
```

其中，path="/test"为虚拟目录名称，docBase="d:\test\"为 Web 服务子目录的物理路径。最后保存并关闭 server.xml 文件，重启 Tomcat。把上一节新建的 Web 项目复制到 d:\test\目录下。

在浏览器的地址栏输入 http://127.0.0.1:8080/test/h/WebRoot/index.jsp，如果出现如图 1-34 所示的页面，即为修改成功。

本 章 小 结

通过本章的学习，应基本了解 JSP 的概念、技术特点、运行原理、JSP、JavaBean 和 Java Servlet 的关系，掌握搭建 JSP 运行环境的方法，以及 JSP 开发工具的安装和使用，为

后续的学习打下基础。

本章习题

一、选择题

1. 当用户请求JSP页面时,JSP引擎就会执行该页面的字节码文件响应客户的请求,执行字节码文件的结果是(　　)。

A. 发送一个JSP源文件到客户端　　B. 发送一个Java文件到客户端

C. 发送一个HTML页面到客户端　　D. 什么都不做

2. 当多个用户请求同一个JSP页面时,Tomcat服务器为每个客户启动一个(　　)。

A. 进程　　B. 线程　　C. 程序　　D. 服务

3. 下列动态网页和静态网页的根本区别描述错误的是(　　)。

A. 静态网页服务器端返回的HTML文件是事先存储好的

B. 动态网页服务器端返回的HTML文件是程序生成的

C. 静态网页文件里只有HTML标记,没有程序代码

D. 动态网页中只有程序,不能有HTML代码

4. (　　)不是JSP运行必需的。

A. 操作系统　　B. JavaJDK

C. 支持Jsp的Web服务器　　D. 数据库

5. URL是Internet中资源的命名机制,URL由(　　)三部分构成。

A. 协议、主机DNS名或IP地址和文件名

B. 主机、DNS名或IP地址和文件名、协议

C. 协议、文件名、主机名

D. 协议、文件名、IP地址

6. 下列说法中(　　)是正确的。

A. Apache用于ASP技术所开发网站的服务器

B. IIS用于CGI技术所开发网站的服务器

C. Tomcat用于JSP技术所开发网站的服务器

D. WebLogic用于PHP技术所开发网站的服务器

7. Tomcat服务器的默认端口号是(　　)。

A. 80　　B. 8080　　C. 21　　D. 2121

二、简答题

1. 为什么要为JDK设置环境变量?

2. Tomcat和JDK是什么关系?

3. 什么是Web服务根目录、子目录、相对目录?如何配置虚拟目录?

情境二

Web 开发基础——HTML 应用

Project 2

本章将为大家介绍 HTML 的结构和常用的标记，这是为了让大家更加了解我们使用网页设计工具生成的页面代码所代表的含义。

本章学习的目的与要求是掌握 HTML 网页中表格标记、文本标记、链接标记等常用标记的结构和写法。

2.1 常用 HTML 标记

实战演练 1——段落标记

【题目】

本题目要求使用段落标记输出三行文字，对齐方式分别为居中、左对齐、右对齐，另外再输出一个标尺线。

【分析】

本例题是对于 HTML 常用标记的一个体验练习，在这个例题中向大家展示一个 HTML 网页的文档结构，并且应用到了标题栏<title>、段落<p>、换行
、标尺线<hr>标记。效果如图 2-1 所示。

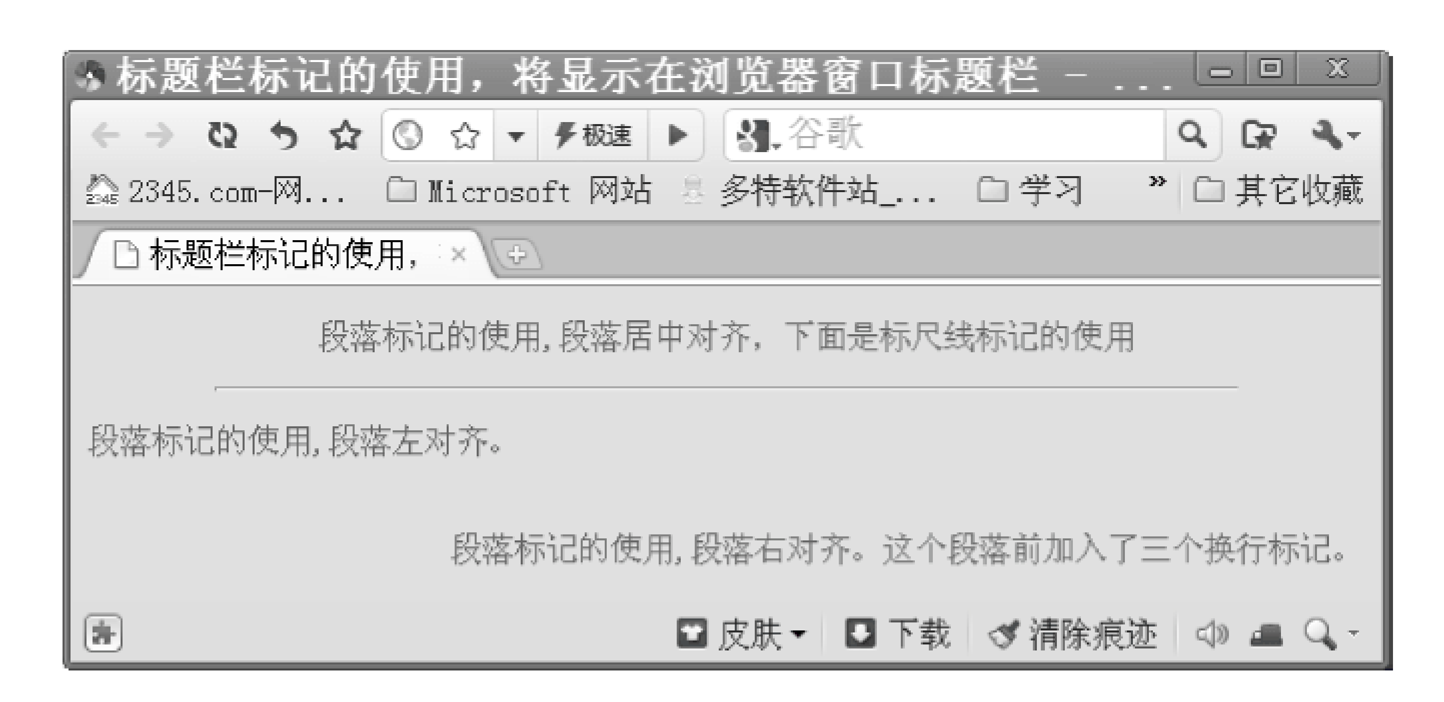


图 2-1 段落标记运行效果网页

【源代码】

```
<html>
<head><title>标题栏标记的使用,将显示在浏览器窗口标题栏</title></head>
<body text="#666666" bgcolor="#FFCCFF">
```

```
<p align="center">段落标记的使用,段落居中对齐,下面是标尺线标记的使用
<hr size="2" width=80%>
<p align="left">段落标记的使用,段落左对齐。
<br /><br /><br />
<p align="right">段落标记的使用,段落右对齐。这个段落前加入了三个换行标记。
</body>
</html>
```

2.1.1 HTML 文件结构

1. HTML 介绍

HTML 是英文 Hyper Text Markup Language 的缩写,即超文本标志语言,所谓超文本,是因为它可以加入图片、声音、动画、影视等内容。用它编写的文件(文档)的扩展名是.html 或.htm,可供浏览器解释浏览。可以使用记事本、Dreamweaver 或 MyEclipse 等编辑工具来编写 HTML 文件。HTML 语言通常使用<标志名></标志名>来表示标志的开始和结束(例如<html></html>标志对),因此在 HTML 文档中这样的标志对都必须是成对使用的。

2. HTML 文档基本结构

一个 HTML 文档由一系列的元素和标签组成。元素名不区分大小写。HTML 用标签来规定元素的属性和它在文件中的位置,HTML 超文本文档分文档头和文档体两部分,在文档头里,对这个文档进行了一些必要的定义,文档体才是要显示的各种文档信息。

(1) <html></html>

<html>标志用于 HTML 文档的最前边,用来标识 HTML 文档的开始。</html>标志放在 HTML 文档的最后边,用来标识 HTML 文档的结束,两个标志必须同时使用。

(2) <head></head>

<head>和</head>构成 HTML 文档的开头部分,在此标志对之间可以使用<title>< /title>、<script></script>等标志对,这些标志对都用于描述 HTML 文档的相关信息,<head></head>标志对之间的内容是不会在浏览器的框内显示出来的。两个标志必须同时使用。

(3) <body></body>

<body></body>是 HTML 文档的主体部分,在此标志对之间可包含<p>、</p>、<h1>、</h1>、
、<hr>等众多的标志,它们所定义的文本、图像等将会在浏览器的框内显示出来。两个标志必须同时使用。

(4) <title></title>

使用过浏览器的人可能都会注意到浏览器窗口最上边蓝色部分显示的文本信息,那些信息一般是网页的“主题”,要将网页的主题显示到浏览器的顶部其实很简单,只要在<title></title>标志对之间加入想要显示的文本即可。注意:<title>< /title>标志对只能放在<head></head>标志对之间。

2.1.2　HTML常用标记

1. HTML的标签与属性

HTML的标签有单标签和成对标签两种。成对标签是由首标签<标签名>和尾标签</标签名>组成的，成对标签的作用域只作用于这对标签中的文档。单独标签<标签名>在相应的位置插入元素就可以了，大多数标签都有自己的一些属性，属性要写在首标签内，属性用于进一步改变显示的效果，各属性之间无先后次序，属性是可选的，属性也可以省略而采用默认值，其格式如下：

<标签名字属性1　属性2　属性3 … >内容</标签名字>

作为一般的原则，大多数属性值不用加双引号。但是包括空格、%、#等特殊字符的属性值必须加入双引号。如：

<body text="#666666" bgcolor="#FFCCFF">

注意：输入始标签时，一定不要在"<"与标签名之间输入多余的空格，也不能在中文输入法状态下输入这些标签及属性，否则浏览器将不能正确地识别括号中的标志命令，从而无法正确地显示你的信息。

2. HTML的主要标记——文字布局

(1) 行的控制

换段标记格式：

<p algin=参数>...</p>

换行标记格式：

不换行标记格式：

<nobr>

<p></p>标志对是用来创建一个段落，在此标志对之间加入的文本将按照段落的格式显示在浏览器上。

(2) 文字对齐

algin属性是标记的对齐属性，通过设置它的值可以对齐标记中的内容，如<p align="center">段落标记的使用，段落居中对齐</p>。另外，还可以通过<center>标记对文本在页面中进行居中显示。

实战演练2——在网页中使用各种文本格式

【题目】

本题目是对于网页中常用文本标记的使用示范。

【分析】

在这个练习中应用到了标题<h1><h2><h3>标记、粗体<b>、斜体<i>、下划线<u>、上标<sub>、下标<sup>等标记。效果如图 2-2 所示。

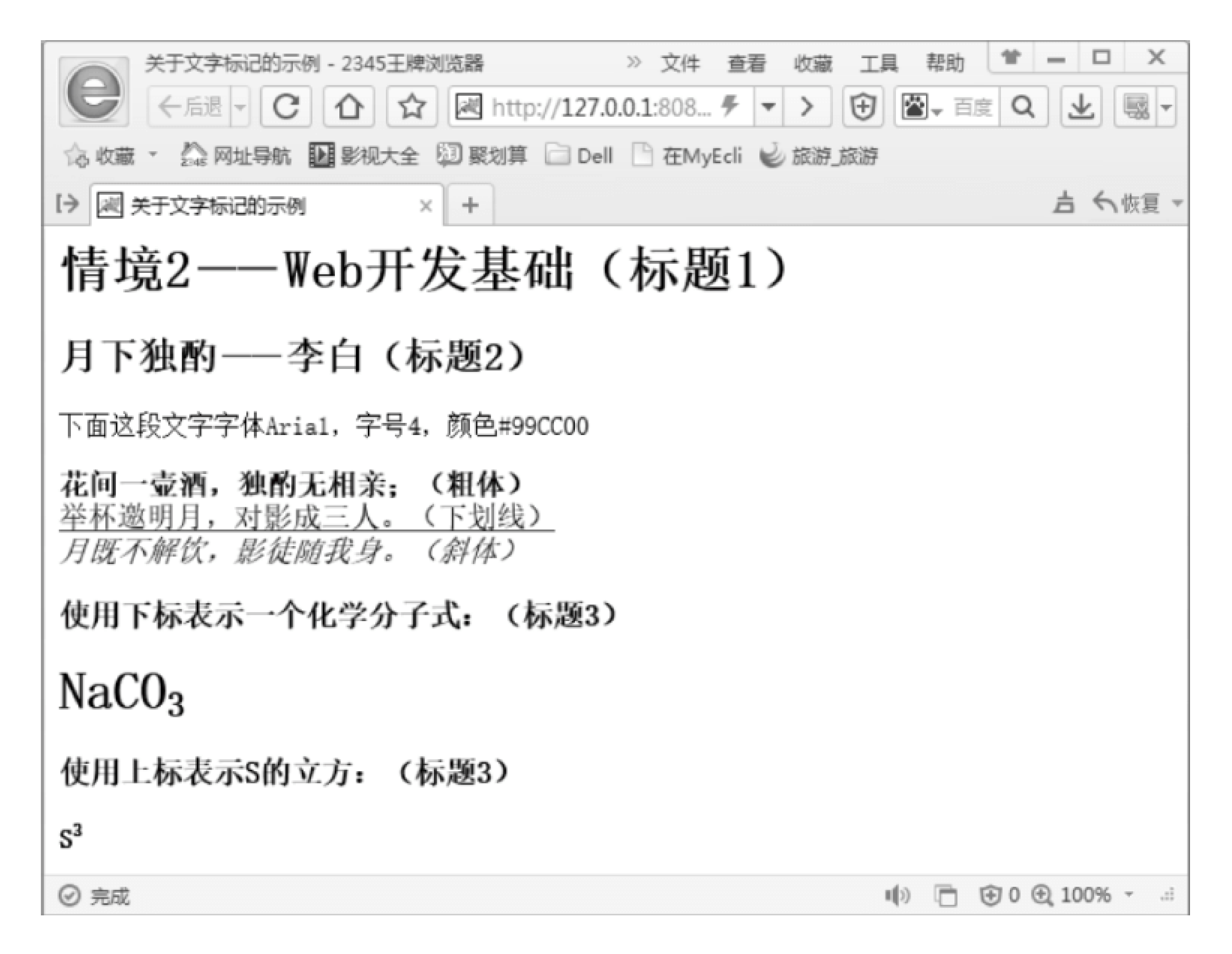


图 2-2　文字标记运行效果网页

【源代码】

```
<html>
<head><title>关于文字标记的示例</title></head>
<body>
<h1>情境 2——Web 开发基础(标题 1)</h1>
<h2>月下独酌——李白(标题 2)</h2>
<font face="Arial" size="4" color="#99CC00" >
    <p>这段文字字体 Arial,字号 4,颜色#99CC00</p>
    <p><b>花间一壶酒,独酌无相亲;(粗体)</b></p>
    <p><u>举杯邀明月,对影成三人。(下划线)</u></p>
    <p><i>月既不解饮,影徒随我身。(斜体)</i></p>
</font>
<h3>使用下标表示一个化学分子式:(标题 3)</h3>
<font size="6">
<p><b>NaCO<sub>3</sub></b></p>
</font>
<h3>使用上标表示 S 的立方:(标题 3)</h3>
<p><b>S<sup>3</sup></b></p>
</body>
</html>
```

2.1.3 HTML的主要标记——字体标记

1．标题字体

标题字体标记的格式：

<h# align=参数>…</h#>其中#=1,2,3,4,5,6

标签用于设置网页中的标题文字，被设置的文字将以黑体或粗体的方式显示在网页中。其参数为 left(左)，enter(中)，right (右)，如上面的例题中用到的，<h2>月下独酌——李白(标题 2)</h2>。

2．字体格式

<FONT>标签用于控制文字的字体、大小和颜色。控制方式是利用属性设置得以实现的，格式如下：

…

face 属性设置文字使用的字体，如果用户的系统中没有 face 属性所指的字体，则将使用默认字体。size 属性的取值为 1～7，也可以用"+"或"－"来设定字号的相对值。color 属性的值为：rgb 颜色 #nnnnnn 或颜色的名称。如上面的例题中用到的，<font face="Arial" size="4" color="#99CC00" >。

3．字体样式

在有关文字的显示中，常常会使用一些特殊的字形或字体来强调、突出、区别以达到提示的效果。

粗体字：<p>花间一壶酒，独酌无相亲；(粗体)</p>

下划线：<p><u>举杯邀明月，对影成三人。(下划线)</u></p>

斜体字：<p><i>月既不解饮，影徒随我身。(斜体)</i></p>

带下标的：<p>NaCO₃</p>

带上标的：<p>S³</p>

还有一些举例：

用于强调的文本，一般显示为斜体字

用于特别强调的文本，显示为粗体字

<cite>用于引证和举例，通常是斜体字</cite>

<code>用来指出这是一组代码</code>

<small>规定文本以小号字显示</small>

<big>规定文本以大号字显示</big>

<samp>显示一段计算机常用的字体，即宽度相等的字体</samp>

<kbd>由用户输入文本，通常显示为较粗的宽体字</kbd>

<var>用来表示变量，通常显示为斜体字</var>

<dfn>表示一个定义或说明，通常显示为斜体字</dfn>

2.2 HTML 的表格应用

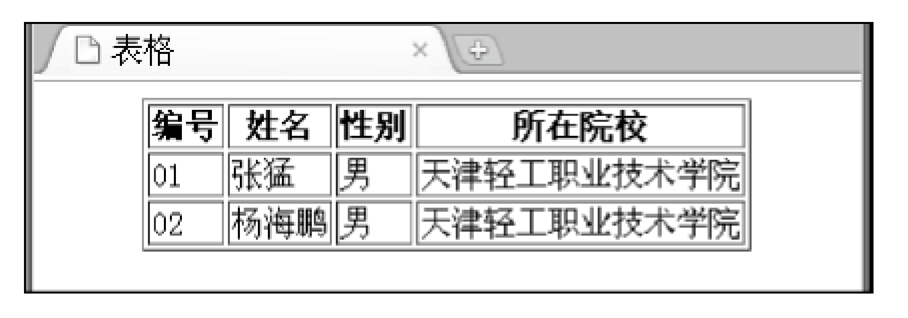
实战演练 1——表格 1

【题目】

本范例介绍如何在网页中建立一个基本表格。

【分析】

在网页制作中表格的使用是非常普遍的，可以说在网页的世界里有了表格才有了规矩。在整个网页中，可以使用表格来划分页面的区块，无论是文字或图片都可以在表格中找到适当的位置摆放。此范例介绍了在网页中表格的基本结构以及使用方法。表格使用<table>表示、标题行使用<th>表示，普通行使用<tr>表示，单元格使用<td>表示。范例运行效果如图 2-3 所示。



编号	姓名	性别	所在院校
01	张猛	男	天津轻工职业技术学院
02	杨海鹏	男	天津轻工职业技术学院

图 2-3 在网页中建立的一个基本表格

【源代码】

```
<html>
<head><title>表格</title></head>
<body>
<center>
<table border="1">
  <tr>
       <th>编号</th>
       <th>姓名</th>
       <th>性别</th>
       <th>所在院校</th>
  </tr>
  <tr>
       <td>01</td>
       <td>张猛</td>
       <td>男</td>
       <td>天津轻工职业技术学院</td>
  </tr>
  <tr>
       <td>02</td>
       <td>杨海鹏</td>
       <td>男</td>
       <td>天津轻工职业技术学院</td>
  </tr>
</table>
</center>
</body>
</html>
```

221 定义表格的基本语法

表格标志对于制作网页是很重要的，主要是因为表格不但可以固定文本或图像的输出，而且还可以任意地进行背景和前景颜色的设置。

定义表格的格式如下：

```
<table>
…
</table>
```

定义表行用<tr></tr>，定义表头用<th></th>，定义单元格用<td></td>。在一个最基本的表格中，必须包含一组<table>标签，一组标签<tr>和一组<td>标签或<th>标签。使用方法如上面例题所示。

222 单元格内文字布局

文字在表格中的布局包括水平方向和垂直方向两种。它们的语法如下所述。

水平方向：

```
<tr align=参数> </tr>
<th align=参数> </th>
<td align=参数> </td>
```

参数＝left、center、right。

垂直方向：

```
<tr valign=参数> </tr>
<th valign=参数> </th>
<td valign=参数> </td>
```

参数＝top、middle、bottom、baseline。

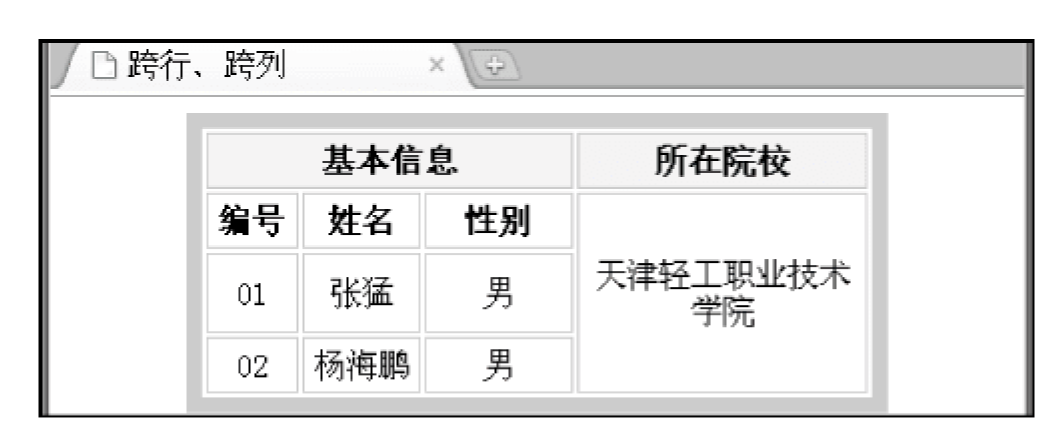
实战演练 2——表格 2

【题目】

本范例介绍如何在表格中实现跨行、跨列的操作。

【分析】

在日常的表格应用当中，跨行或跨列的操作非常普遍，那么在网页中要如何完成这样的操作呢？实现跨行的方法是在行的属性中加入 colspan；实现跨列的方法是在列的属性中加入 rowspan。本题目运行效果如图 2-4 所示。



基本信息			所在院校
编号	姓名	性别	天津轻工职业技术学院
01	张猛	男	
02	杨海鹏	男	

图 2-4 跨行、跨列操作示例

【源代码】

```
<html>
<head><title>跨行、跨列</title></head>
<body>
<table width="350"  bordercolor="#CCCCCC" heigth="290" border="8" align="center" cellpadding="4" cells5  pacing="2">
  <tr >
    <th colspan="3" bgcolor="#FFFF99">基本信息</th>
    <th width="132" align="center" valign="middle" bgcolor="#FFFF99">所在院校</th>
  </tr>
  <tr >
    <th width="36">编号</th>
    <th width="49">姓名</th>
    <th width="63">性别</th>
    <td rowspan="3" align="center" valign="middle">天津轻工职业技术学院</td>
  </tr>
  <tr >
    <td height="30" align="center" valign="middle">01</td>
    <td align="center" valign="middle" >张猛</td>
    <td align="center" valign="middle">男</td>
  </tr>
  <tr >
    <td align="center" valign="middle">02</td>
    <td align="center" valign="middle">杨海鹏</td>
    <td align="center" valign="middle">男</td>
  </tr>
</table>
</body>
</html>
```

223 跨多行、多列的单元格

要创建跨多行、多列的单元格，只需在<th>或<td>中加入 rowspan 或 colspan 属性的属性值，默认值为 1，该属性值表明了表格中要跨越的行或列的个数。

跨多列的语法：

```
<th colspan=#>  <td colspan=#>
```

colspan 表示跨越的列数，如 colspan=2 表示这一格的宽度为两个列的宽度。

跨多行的语法：

```
<th rowspan=#>  <td rowspan=#>
```

rowspan 表示跨越的行数，如 rowspan=2 表示这一格跨越表格两个行的高度。

224 嵌套表格

在 HTML 页面中，使用表格排版是通过嵌套来完成的，即一个表格内部可以嵌套另一个表格。用表格来排版页面的思路是：由总表格规划整体的结构，由嵌套的表格负责

各个子栏目的排版，并插入表格的相应位置，这样就可以使页面的各个部分有条不紊、互不冲突、清晰整洁。在实际做网页时一般不显示边框，边框的显示可根据自己的爱好来设定。

2.3　在 HTML 中插入图像和超级链接

实战演练——超级链接

【题目】

本范例介绍网页中图片的添加和超级链接的使用。

【分析】

在此例题中使用到了链接标记<a href>，以此方法链接到了前面做过的例题网页，以及本门课程精品课网站的首页，还有邮件链接的方法。在这个网页中我们还使用了<img/>标记添加了图片。题目运行效果如图 2-5 所示。

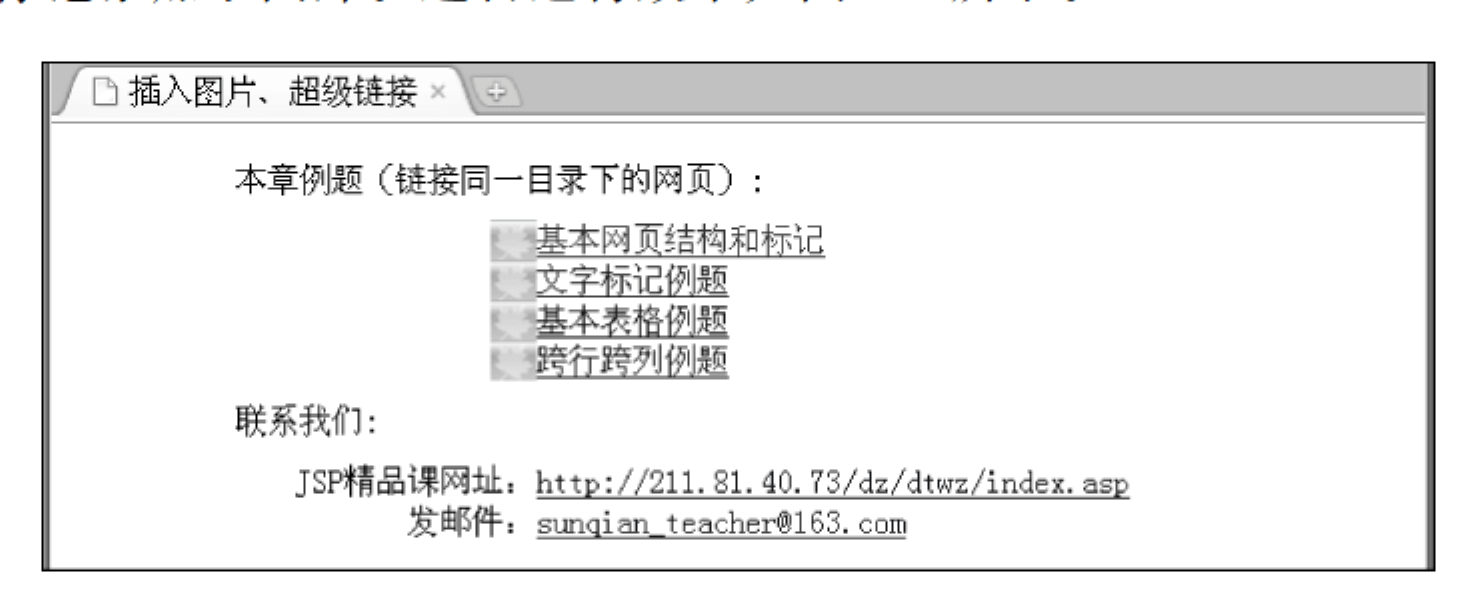


图 2-5　插入图片和超级链接示例

【源代码】

```
<html>
<head><title>插入图片、超级链接</title></head>
<body>
<table width="500" border="0" align="center" cellpadding="0" cellspacing="0">
  <tr>
    <td height="40" colspan="2" align="left" valign="middle">
    本章例题(链接同一目录下的网页): </td>
  </tr>
  <tr>
    <td width="150" align="right"><img src="sub05_07.gif" width="23" height="20" />
    </td>
    <td width="350" height="20">
          <a href="ch2-2.html" title="" target="_blank">基本网页结构和标记</a>
          </td>
  </tr>
  <tr>
    <td align="right"><img src="sub05_07.gif" width="23" height="20" /></td>
    <td height="20"><a href="ch2-3.html" title="" target="_blank" >文字标记例题
    </a></td>
```

```
</tr>
<tr>
  <td align="right"><img src="sub05_07.gif" width="23" height="20" /></td>
  <td height="20"><a href="ch2-6.html" title="" target="_blank" >基本表格例题
  </a></td>
</tr>
<tr>
  <td align="right"><img src="sub05_07.gif" width="23" height="20" /></td>
  <td height="20"><a href="ch2-9.html" title="" target="_blank" >跨行跨列例题
  </a></td>
</tr>
<tr>
  <td height="40" colspan="2" align="left" valign="middle">联系我们:</td>
</tr>
<tr>
  <td align="right">JSP 精品课网址:</td>
  <td height="20"><a href="http://211.81.40.73/dz/dtwz/index.asp"title="" target=
  "_blank"> http://211.81.40.73/dz/dtwz/index.asp </a></td>
</tr>
<tr>
  <td align="right">发邮件:</td>
  <td height="20">
  <a href="mailto:sunqian_teacher@163.com" title=" " >sunqian_teacher@163.com</a>
  </td>
</tr>
</table>
</body>
</html>
```

2.3.1 <img> 表单标记

在网页中是通过<img>标记的属性 src 将图像插入网页中的,属性 src 的值可以是路径,也可以是网址。实际上就是通过路径将图形文件嵌入文档中。

插入图像的格式为

```
<img src=参数 1 alt =参数 2>
```

属性 alt 设置浏览器尚未完全读入图像时,在图像位置显示的文字。

设置图像和文字对齐的格式为

```
<img align=参数>
```

其中参数=top,middle,bottom,left,right。

2.3.2 <a href> 标记

1. 建立超级链接

HTML 文件中最重要的应用之一就是超级链接,超级链接是一个网站的灵魂。超文

本具有的链接能力,可层层链接相关文件,这种具有超级链接能力的操作,即称为超级链接。

超级链接格式为

...

说明:标签<a>表示一个链接的开始,表示链接的结束;

href:定义了这个链接所指的目标地址,目标地址是最重要的,一旦路径上出现差错,该资源就无法访问;

title:该属性用于指定指向链接时所显示的标题文字;

target:该属性用于指定打开链接的目标窗口,其默认方式是原窗口。该属性的属性值,如表 2-1 所示。

表 2-1 target 的属性值

属性值	描 述
_parent	在上一级窗口中打开,一般使用分帧的框架页会经常使用
_blank	在新窗口打开
_self	在同一个帧或窗口中打开,这项一般不用设置
_top	在浏览器的整个窗口中打开,忽略任何框架

2. 链接路径

标记<a>的 href 属性可以接受绝对路径和相对路径。可以通过绝对路径链接站点外的资源,如JSP 精品课网站;也可以通过相对路径链接站点内资源,如基本表格例题。

3. 发送 E-mail

在 HTML 页面中,可以建立 E-mail 链接。当浏览者单击链接后,系统会启动默认的本地邮件服务系统发送邮件。

格式为

...

如上题中用到的。

4. 表格标题

设置表格标题的格式为

<caption align=参数></caption>

其中参数=left、center、right;

<caption valign=参数></caption>

其中参数＝top、bottom。

注意：<caption>应放在<table>标签内，在表格行标签<tr>标签之前。<caption>标签的默认属性是标题位于表格的上方中间位置。

本 章 小 结

本章大家学习了文本标记、表格标记、链接标记、图片标记等 HTML 的基础知识，了解网页的基本结构是我们顺利进入动态网站开发殿堂的基石。

本 章 习 题

【思考题】 在基本表格中设置一些表格属性，完成如图 2-6 所示的页面效果。



编号	姓名	性别	所在院校
01	张猛	男	天津轻工职业技术学院
02	杨海鹏	男	天津轻工职业技术学院

图 2-6 基本表格页面效果

情境三

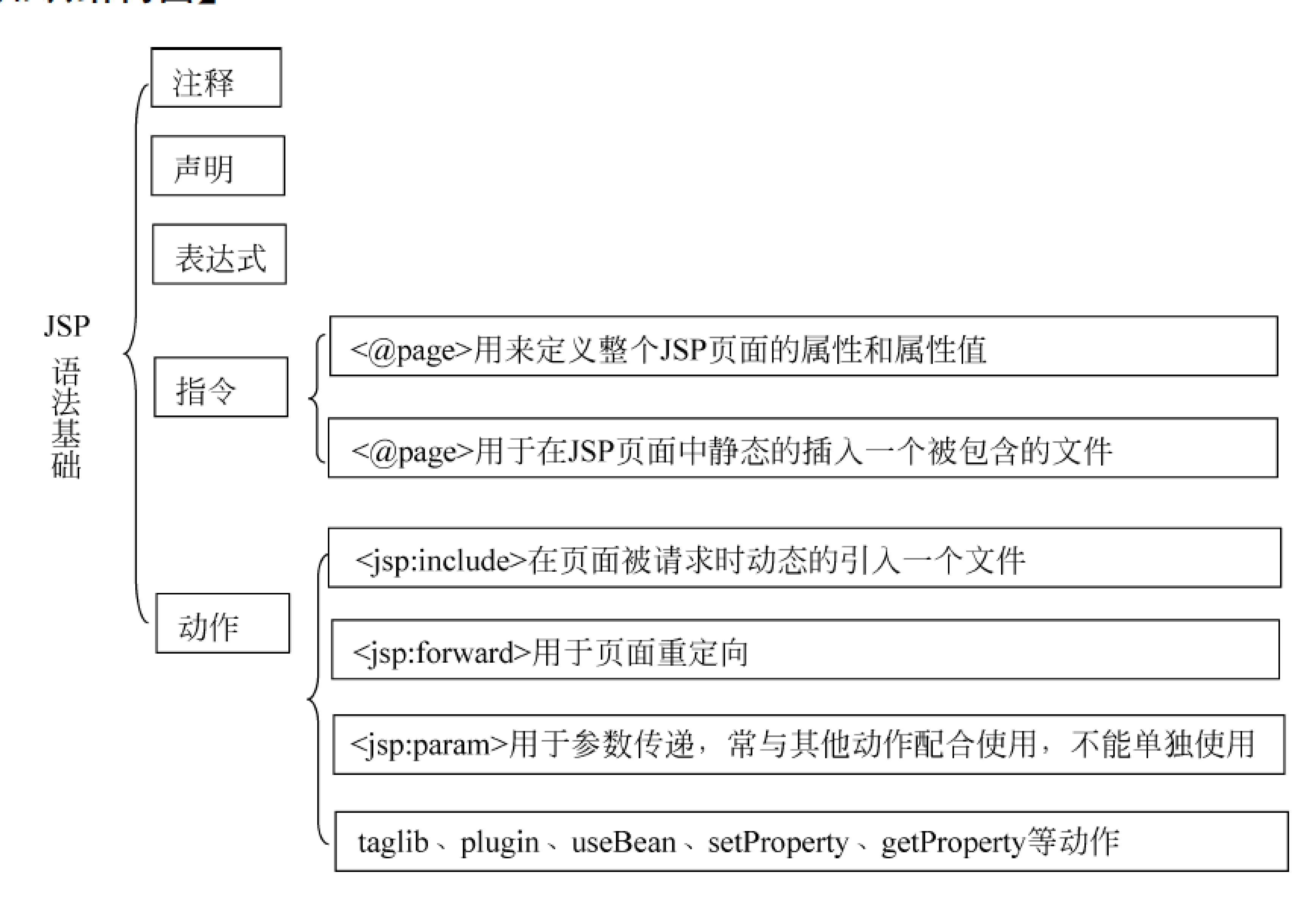
JSP 语法基础

Project 3

【学习目标】

1. 掌握 JSP 注释、JSP 指令、脚本元素及 JSP 动作的使用方法。
2. 能够根据项目需要，熟练使用 JSP 语法进行项目开发。

【知识结构图】



3.1 JSP 注释、表达式

实战演练 1——JSP 页面基本结构体验

【题目】

数字 1～5 的求和。

【分析】

本题目主要是为了让大家体验 JSP 页面的基本结构，以及两种注释的书写方法。

JSP 页面可以由 7 种元素组成，分别是：普通的 HTML 标记、JSP 指令标记、JSP 动作标记、变量声明与方法声明、程序片、表达式、注释。

本题目中用到了 JSP 注释和 HTML 注释。

【运行效果】

在开发工具 MyEclipse 中新建 Web Project，工程名称命名为 jsp-yufa（如图 3-1 所示），在此工程中的 WebRoot 文件夹下新建工程文件 ex4.jsp。程序输入完毕并且保存之后，运行开发工具 MyEclipse 中的 Tomcat 服务器（如图 3-2 所示），并且选择服务器需要服务的工程（如图 3-3(a)和图 3-3(b)所示）。最后在浏览器中输入地址 http://127.0.0.1:8080/jsp-yufa/ex4.jsp 就可以看到运行结果了（如图 3-4 所示）。

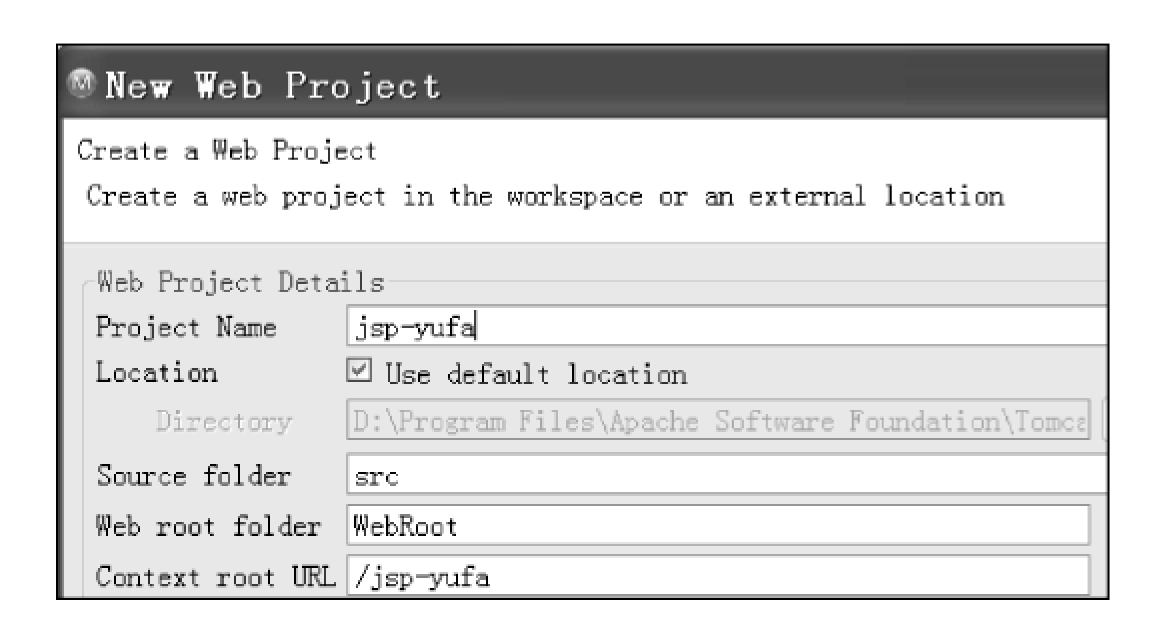


图 3-1 工程名称为 jsp-yufa

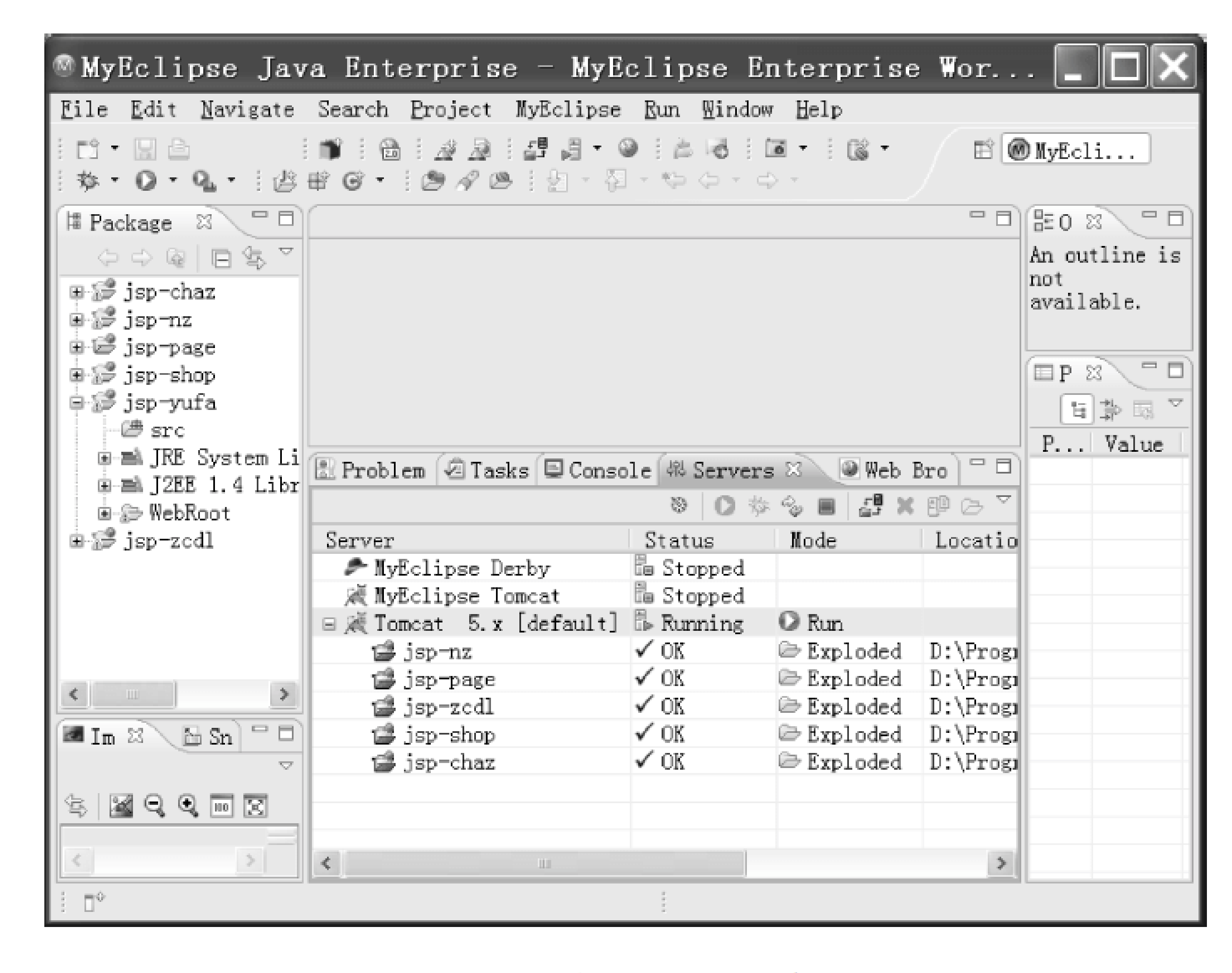
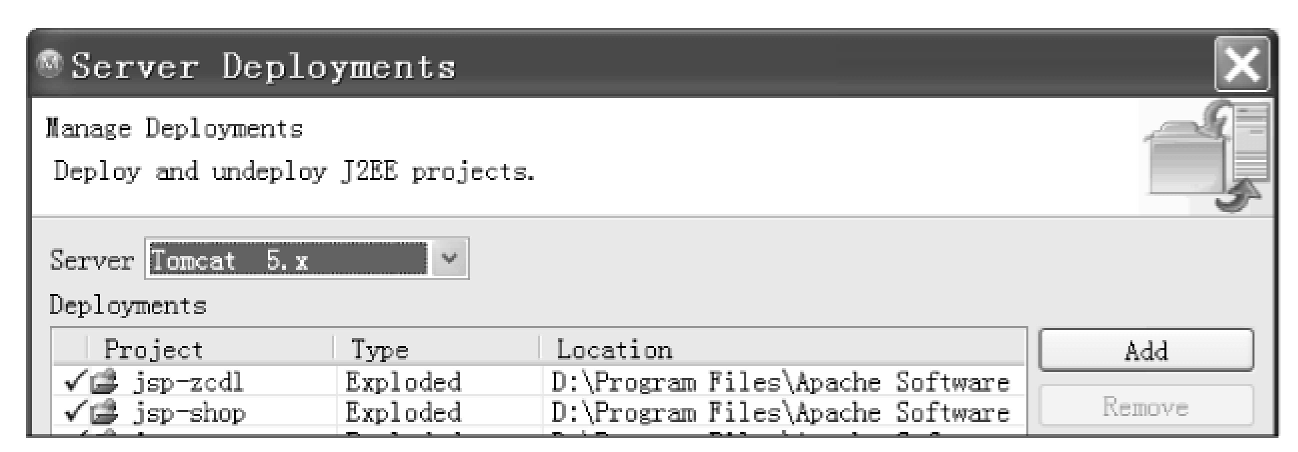


图 3-2 运行 Tomcat 服务器

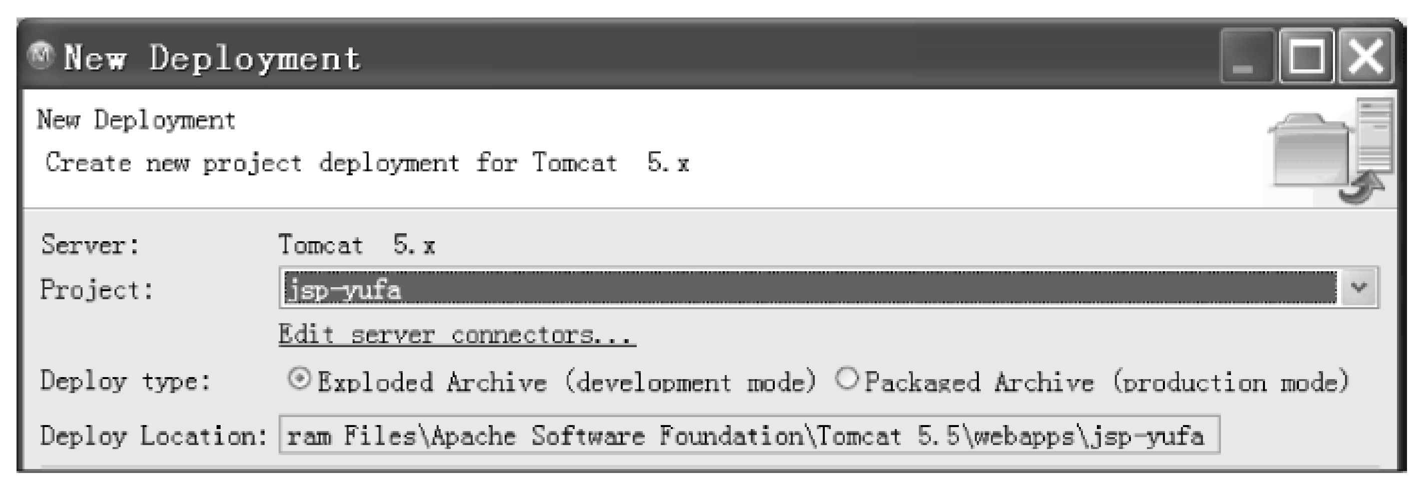
【源代码】

【ex4.jsp】

```
<%@ page contentType="text/html; charset=gb2312" language="java"%>
<html>
  <head>
    <title>JSP 页面结构体验</title>
  </head>
```



(a)



(b)

图 3-3　添加需要服务的工程

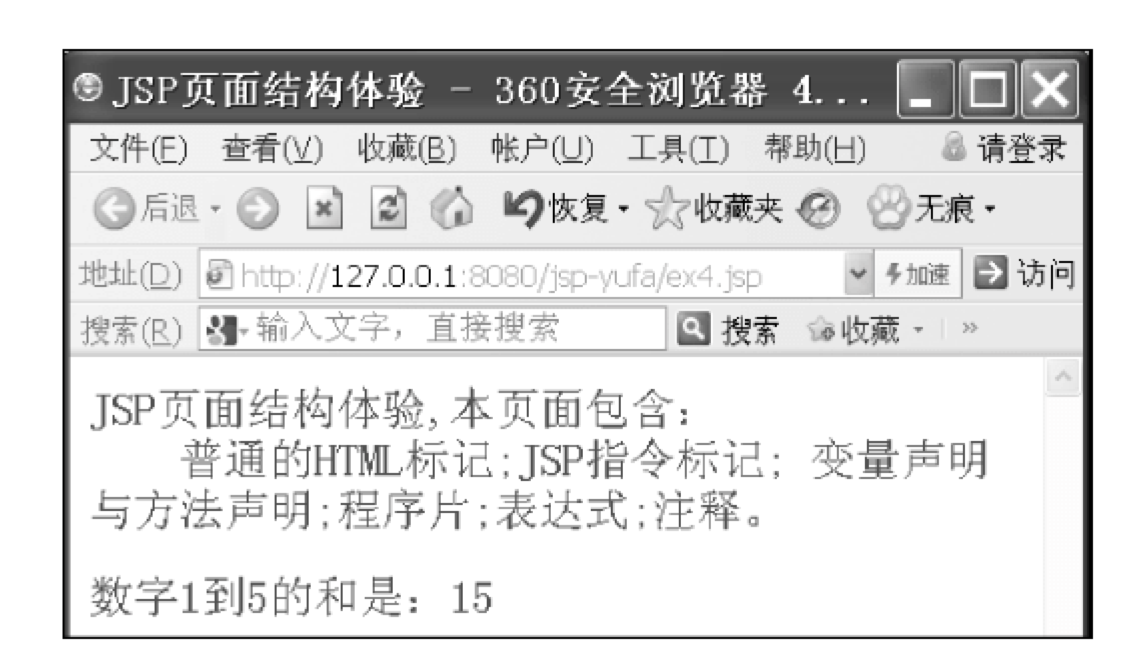


图 3-4　ex4.jsp

```
<body>
<!-- 这是一个 HTML 注释,下面是一个方法声明,add 方法用于完成加法运算 -->
<%-- 这是一个 JSP 注释,下面是一个方法声明,add 方法用于完成加法运算 --%>
<%!//声明 add 方法,带入两个 int 型参数进行运算,返回值为 int 类型
public int add(int i,int j)
{
  int temp=0;
  for(;i<=j;i++)
  {     temp+=i;        }
  return temp;
}
%>
<font size=5 color=#0066cc>
  <p>JSP 页面结构体验,本页面包含:
  <br/>    
  普通的 HTML 标记;JSP 指令标记;变量声明与方法声明;程序片;表达式;注释。
  <% //调用 add 方法,传递的参数为 1 和 5
```

```
23            out.println("<p>数字 1 到 5 的和是: "+add(1,5));
24        %>
25        </font>
26    </body>
27 </html>
```

【补充说明】

上面源代码中第 1 行是 JSP 指令标记，以<%@开头，以%>结束；第 2 行到第 6 行是 HTML 标记；第 7、8 行是注释；第 9 行到第 17 行是方法声明，以<%！开头，以%>结束；第 22 行到第 24 行是 Java 程序片，以<%开头，以%>结束；第 22 行，以//开头的语句是 Java 注释。

3.1.1 JSP 页面结构

JSP 页面由 7 种元素组成。

(1) 普通的 HTML 标记，如< html >、< head >、< body >等。

(2) JSP 指令标记，如<@ page >、< @ include >等。

(3) JSP 动作标记，以<jsp:开头，以/>或</jsp:动作>结束的标记。

(4) 变量声明与方法声明(Declaration)：在程序中<%和%>之间声明的变量只在本页面有效，是局部变量，在不同的客户端访问同一个页面时，此变量不能共享；但如果在<%！和%>之间声明变量就可以在不同页面共享，其有效范围是整个 JSP 页面，在服务器关闭后变量才会被释放。这在后面的很多例题中都将体现。

(5) 程序片：在 JSP 中，在<%和%>之间书写的程序代码称为 Java 程序片，一个 JSP 页面可以有多个 Java 程序片。

(6) 表达式：用<%=和%>可以直接输出变量或表达式的值，变量或表达式的值将作为一个字符串在浏览器中输出。这种方法在 JSP 编程中比较常用，特别在 Java 和 HTML 标记混合编写 JSP 程序中是较为常用的。

(7) 注释，包含 HTML 注释和 JSP 注释。

3.1.2 注释

适当地在 JSP 程序中加入注释可以增加程序的可读性，以方便维护。即使对于程序编写人员，注释对调试和编写程序也可起到良好的辅助作用。程序员在程序中书写注释是一种良好的习惯，在平时的学习和实践中要注意培养这种习惯。

JSP 文件中注释可分为两种：HTML 注释和 JSP 注释。

1. HTML 注释

在符号<!--与-->之间的是 HTML 注释，JSP 不会解释 HTML 注释，而直接把它交给客户端的浏览器。因此，这种注释在浏览器中直接查看原文件时可以看到。

其格式如下：

```
<!- -注释内容- ->
```

用法,如上面题目中用到的:

<!-- 这是一个 HTML 注释,下面是一个方法声明,add 方法用于完成加法运算 -->

2. JSP 注释

JSP 注释的方法有以下 4 种。格式分别如下:

(1) <%- -注释内容- -%>

(2) / /注释内容

(3) / * 注释内容 * /

(4) / ** 注释内容 * /

第一种注释会被 JSP 引擎忽略,一般用来对 Java 程序片进行说明。如上面题目中用到的:<%--这是一个 JSP 注释,下面是一个方法声明,add 方法用于完成加法运算--%>。

第二种注释是单行注释,如上面题目中用到的://调用 add 方法,传递的参数为 1 和 5。

第三种方式可以是单行注释,也可以是多行注释。

第四种方式是 Java 所特有的 doc 注释。

注释在编写程序和调试程序中特别有用,比如在调试一段程序中的几行程序时,可以注释其他程序,这样 JSP 引擎就不会编译它,这是一种比较常用的程序调试方法。JSP 注释是不会在浏览器中通过查看源文件的方法看到的。

实战演练 2——字符串作为表达式输出

【题目】

题目完成的功能是把字符串"表达式练习",作为表达式进行输出操作。

【分析】

本题目是对表达式的练习,程序中把表达式放入一个 for 循环中,做固定次数的输出。

【运行效果】

打开开发工具 MyEclipse,在上一个题目所建立的工程 jsp-yufa 中的 WebRoot 文件夹下,新建工程文件 ex2.jsp。程序输入完毕并且保存之后,运行开发工具 MyEclipse 中的 Tomcat 服务器。最后在浏览器中输入地址 http://127.0.0.1:8080/jsp-yufa/ex2.jsp 就可以看到运行结果了(如图 3-5 所示)。



图 3-5　表达式练习的运行结果

【源代码】

【ex2.jsp】

```
<%@ page contentType="text/html; charset=gb2312" language="java"%>
<html>
```

```
3      <head>
4        <title>表达式练习</title>
5      </head>
6      <body>
7      <center>
8      <font color=#006600 size=5>
9         <%
10        String s="表达式练习";
11        int i=0;
12        for(;i<=6;i++)
13        {
14            %>
15            <%=s %>
16            <%
17            out.println("<br>");
18          }
19          %>
20     </font>
21     </center>
22     </body>
23   </html>
```

【补充说明】

第 7 行是为了把输出的文本居中显示；第 8 行是设置输出的文本字号和颜色；第 9 行到第 19 行是两段程序片和一个表达式，使用 for 循环把字符串 s 作为表达式做循环输出。

3.1.3 表达式

JSP 表达式的语法格式如下：

<%=表达式%>

在<%=和%>标记之间放置 Java 表达式，可以直接输出 Java 表达式的值。例如上面题目中的用法：<%=s %>。表达式的值由服务器负责计算，并将计算值转换成字符串发送给客户端显示。表达式在 JSP 编程中较常用，特别是在与 HTML 标记混合编写时使用较多。

3.2 JSP 的变量声明、方法声明

实战演练 1——判断并计算三角形面积

【题目】

在界面的三个文本框中输入数字作为三角形的三个边，然后提交这三个数字。如果这三个数字能构成三角形，则计算三角形的面积；如果不能构成三角形，则输出文字“不能构成三角形”。在显示计算结果的时候要显示刚刚输入过的三个数字。最后，要有一个“返回”的链接，能够回到输入界面。

【分析】

本题目以判断三角形和计算三角形的面积为例，向大家介绍 JSP 脚本和声明的用法。代码中使用了<%！和%>标记声明变量。这是一个让大家熟悉 JSP 程序书写方法的过渡阶段。

【运行效果】

打开开发工具 MyEclipse，在工程 jsp-yufa 中的 WebRoot 文件夹下，新建工程文件 ex1_1.jsp 和 ex1_2.jsp。程序输入完毕并且保存之后，运行开发工具 MyEclipse 中的 Tomcat 服务器。在浏览器中输入地址 http://127.0.0.1:8080/jsp-yufa/ex1_1.jsp。

（1）首先运行的是 ex1_1.jsp 页面，效果如图 3-6 所示。在文本框中分别输入三个数字，然后单击“计算”按钮，页面把已输入的三个数值提交到 ex1_2.jsp 页面。

（2）ex1_2.jsp 页面运行效果如图 3-7 所示，显示了上一页面输入的三个数值，并且显示运算结果，另外还有“返回”链接可以回到上一个页面继续操作。



图 3-6　ex1_1.jsp 页面



图 3-7　ex1_2.jsp 页面

【源代码】

【ex1_1.jsp】 本页面的设计中使用了一个表单，表单中放置了三个文本框，名称分别为 a、b、c，还有一个提交按钮。

```
<%@ page contentType="text/HTML; charset=gb2312" language="java"%>
<HTML>
  <head>
    <title>JSP 脚本、声明</title>
  </head>
  <body bgcolor="#FFFFCC">
  <center>
    <font color="#946714" size="3">
    <p>请输入三角形三个边的长度：</p>
    <form  action="ex1_2.jsp" method="post" name="form">
        <input type="text" name="a" size="5"/>
        <input type="text" name="b" size="5"/>
        <input type="text" name="c" size="5"/>
        <input type="submit" value="计算" name="submit" />
    </form>
  </center>
  </body>
</HTML>
```

【ex1_2.jsp】

```
1  <%@ page contentType="text/HTML; charset=gb2312" language="java"%>
2  <HTML>
3    <head>
4      <title>JSP 脚本、声明</title>
5    </head>
6    <body bgcolor="#FFFFCC">
7    <center>
8      <font color="#946714" size="3">
9      <%!
10         double s=0;
11         String answer=null;
12     %>
13     <%
14         double a=Double.parseDouble(request.getParameter("a"));
15         double b=Double.parseDouble(request.getParameter("b"));
16         double c=Double.parseDouble(request.getParameter("c"));
17         if((a+b>c)&&(b+c>a)&&(a+c>b))
18         {
19             double p=(a+b+c)/2;
20             s=Math.sqrt(p * (p-a) * (p-b) * (p-c));
21             out.print("面积是: "+s);
22         }
23         else
24         {
25             answer="您输入的三边不能构成一个三角形";
26             out.print("<br>"+answer);
27         }
28     %>
29     <p>您输入的三边是:
30     <%=a%>   <%=b%>   <%=c%></p>
31     <a href="ex1_1.jsp" target="_self">返回</a>
32   </center>
33   </body>
34 </HTML>
```

【补充说明】

ex1_1.jsp 代码说明

第 10 行到第 15 行是一个表单,以<form>开始,以</form>结束,表单中有三个文本框,文本框的名字分别为 a、b、c,还有一个提交按钮,按钮上的文本为“计算”,当单击“计算”按钮之后,文本框中的数据提交到 ex1_2.jsp 页面。

ex1_2.jsp 代码说明

第 9 行到第 12 行是变量声明,以<%! >开始,以<%>结束;第 13 行到第 28 行是 Java 程序片;第 14 到第 16 行是接收表单 ex1_1.jsp 页面传递过来的数据 a、b、c,并且把数据转换成为 double 类型;第 17 行到第 27 行是一个条件语句,用来判断这三个数值能否构成三角形,如果能构成三角形,则计算三角形的面积,如果不能构成三角形,则显示

文字“您输入的三边不能构成一个三角形”；第 30 行把三个变量做了表达式的输出；第 31 行设置了一个“返回”链接，可以返回到 ex1_1.jsp 页面，继续操作。

3.2.1　使用声明

变量和方法在<%!和%>标记之间声明，声明变量和方法的语法和格式同 Java 语言。

语法格式如下：

```
<%!声明 1;[声明 2;]…%>
```

例如上面题目中用到的：

```
<%!
    double s=0;
    String answer=null;
%>
```

声明变量：就是在<%!和%>标记之间放置 Java 的变量声明语句。变量的数据类型可以是 Java 的任何数据类型。

声明方法：声明的方法在整个页面内有效，称为页面的成员方法。

方法内声明的变量称为局部变量，只在方法内有效，方法调用为其分配空间，调用完毕后释放变量空间；全局变量在所有客户之间该页面有效。

在程序中<%和%>之间声明的变量只在本页面有效，是局部变量，在不同的客户端访问同一个页面时，此变量不能共享。但如果在<%! 和%>之间声明的变量就可以在不同页面共享，其有效范围是整个 JSP 页面，在服务器关闭后变量才会被释放。

3.2.2　表单的设计

在上面题目中首次用到了表单，表单在 Web 网页中用来给访问者填写信息，从而能采集客户端信息，使网页具有交互的功能。一般是将表单设计在一个 HTML 文档中，当用户填写完信息后做提交(submit)操作，于是表单的内容就从客户端的浏览器传送到服务器上，经过服务器处理程序处理后，再将用户所需信息传送回客户端的浏览器上，这样网页就具有了交互性。下面详细了解一下表单在 JSP 中如何使用。

1. 表单标记

一个表单用<form></form>标志来创建。也即定义表单的开始和结束位置，在开始和结束标志之间的一切定义都属于表单的内容。<form>标志具有 action、method 和 target 属性。

(1) action 的值是处理程序的程序名(包括网络路径、网址或相对路径)，如<form action="ex1_2.jsp" method="post" name="form">。

如果这个属性是空值，如<form action=" " method="post" name="form">。

则当前文档的 url 将被使用。当用户提交表单时，服务器将执行当前文档里面的

程序。

(2) method 属性用来定义处理程序从表单中获得信息的方式，可取值为 GET 和 POST 中的一个。GET 方式是处理程序从当前 HTML 文档中获取数据，然而这种方式传送的数据量是有限制的，一般限制在 1KB(255 个字节)以下。POST 方式传送的数据比较大，它是当前的 HTML 文档把数据传送给处理程序，传送的数据量要比使用 GET 方式传送的数据量大得多。

在 JSP 中，服务器端程序与客户端交互最常用的方法就是采用表单提交数据。表单提交的方法主要有两种，一种是 get 方法；另一种是 post 方法。两者最大的区别是：使用 get 方法提供的数据会显示在浏览器的地址栏中；而使用 post 方法则不会显示，故 post 更为常用。

(3) target 属性用来指定目标窗口或目标帧。

2. <input>标记

在 HTML 语言中，标记<input>具有重要的地位，它能够将浏览器中的控件加载到 HTML 文档中，该标记是单个标记，没有结束标记。

写入标记格式如下：

```
<input type=参数 name=参数 ...>
```

例题中的用法：

```
<input type="text" name="c" size="5"/>
<input type="submit" value="计算" name="submit" />
```

Type 属性：指定控件的类型，取值为 text、radio、checkbox、password、hidden、submit、reset 等。

Name 属性：指定空间对象名称，通过该属性，使用 request.getParameter 方法获得用户提交的数据。

定义一个用户输入区，用户可在其中输入信息。此标志必须放在<form></form>标志对之间。<input type="">标志中共提供了 9 种类型的输入区域，本例题中用到的是文本输入区和提交按钮。

实战演练 2——计算圆面积和周长

【题目】

界面上有一个文本框，输入圆的半径，然后单击“计算”按钮，在本页面得出计算的面积和周长。

【分析】

(1) 本题使用了一个 Circle 类，类中有计算面积的方法、有计算周长的方法、有 Circle 对象的初始化。题目是通过创建 Circle 类的对象，用对象调用类的方法来完成运算的。

(2) 页面中需要有一个表单，表单中包括一个文本框和一个提交按钮。

(3) 因为题目只用一个页面完成，所以表单提交到本页面进行运算，当页面第一次被

加载的时候，给半径赋值为“1”；当页面不是首次加载的时候，把表单提交的半径用于后面的运算。

【运行效果】

在工程 jsp-yufa 中的 WebRoot 文件夹下，新建工程文件 ex5. jsp。程序输入完毕并且保存之后，运行开发工具 MyEclipse 中的 Tomcat 服务器。在浏览器中输入地址 http://127. 0. 0. 1:8080/jsp-yufa/ex5. jsp 就可以看到运行结果了（如图 3-8 所示）。



图 3-8　ex5. jsp 页面

【源代码】

【ex5. jsp】

```
<%@ page contentType="text/html; charset=gb2312" language="java"%>
<%! public class Circle
  {
      double r;
      Circle(double r)
      {this.r=r;   }
      //面积
      double mianji()
      {   return Math.PI * r * r;   }
      //周长
      double zhouchang()
      {   return Math.PI * 2 * r;   }
  }
%>
<html>
  <head><title>方法声明练习</title>
  </head>
  <body>
    <font color=#00aaww size=4>
    <p>请输入圆的半径：
    <br>
    <form action="" method=post name=form>
      <input type="text" name="R" vale="1">
      <input type="submit" value="计算" name=submit>
    </form>
    <%
      String R=request.getParameter("R");
      double r;
      if(R!=null)
          {r=Double.parseDouble(R);}
      else
          {r=1;}
      Circle circle=new Circle(r);          //创建对象
```

```
34     %>
35       <p>圆的半径是:
36         <br>
37         <%=r%>
38       <p>圆的面积是:
39         <br>
40         <%=circle.mianji()%>
41       <p>圆的周长是:
42         <br>
43         <%=circle.zhouchang() %>
44     </font>
45   </body>
46 </html>
```

【补充说明】

第 2 行到第 14 行是公共类 Circle 的声明,里面包含了构造方法,用于完成 Circle 对象的初始化,有计算面积的方法 mianji(),有计算周长的方法 zhouchang();第 22 行到第 25 行是 form 表单,表单中有一个文本框,变量名为 R,可以输入圆的半径,初始值设为 1,有一个提交按钮,按钮上的文字设置为计算;第 26 行到第 34 行是 Java 程序片,这一段中第 27 行完成的是接收数据,第 29 到第 32 行的 if 条件语句是用于判断页面是否第一次加载,如果是第一次加载就为圆的半径 r 赋值为 1,如果不是第一次加载就做数据类型转换;第 33 行是创建一个 Circle 类的对象 circle,并且用 r 的值为这个 circle 对象做初始化赋值;第 37 行、40 行、43 行都是表达式的输出,其中第 40 行和第 43 行是使用 circle 对象调用了计算面积和周长的方法得到输出值的。

实战演练 3——可累计的投票

【题目】

统计候选人被投票的数目。

【分析】

(1) 假设有三个候选人,在界面上放置一个表单,其中包括 3 个单选按钮,3 个候选人分别使用不同的标识;一个提交按钮,用于投递选票。

(2) 当选票提交以后,通过候选人标识进行比较,是谁的选票就给谁累计票数。

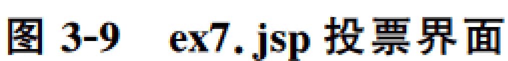
(3) 定义一个页面级数组变量 sum,每个数组元素表示 1 个候选人。

(4) 用 Synchronized 来解决多个线程同时访问同一个共享变量的问题,当一个用户用 Synchronized 操作一个共享变量时,其他线程就必须等待,直到该线程执行完成。

【运行效果】

在工程 jsp-yufa 中的 WebRoot 文件夹下,新建工程文件 ex7.jsp 和 ex7_1.jsp。程序输入完毕并且保存之后,运行开发工具 MyEclipse 中的 Tomcat 服务器。在浏览器中输入地址 http://127.0.0.1:8080/jsp-yufa/ex7.jsp 就可以看到运行结果了(如图 3-9 和图 3-10 所示)。




图 3-9 ex7.jsp 投票界面

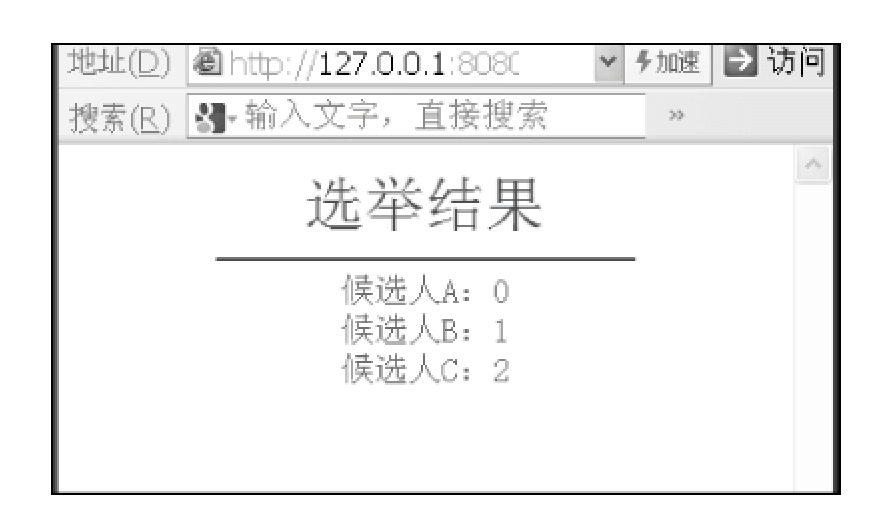


图 3-10 ex7_1.jsp 投票结果

【源代码】

【ex7.jsp】

```
<%@ page contentType="text/html; charset=gb2312" language="java" %>
<html>
  <head>
    <title>表单练习--选举</title>
  </head>
  <body>
    <form method="post" action="ex7_1.jsp">
      <center>
        <font color=#ffggkk size=6>选举</font>
        <hr color="#ffggkk" size="2" width="60%"></hr>
        <font color=#996611 size=4>
          <p>
          <input type="radio" value="A" name="num" />候选人 A<br />
          <input type="radio" value="B" name="num" />候选人 B<br />
          <input type="radio" value="C" name="num" />候选人 C<br /><br />
          <input type="submit" value="投票" name="tou" />
        </font>
      </center>
    </form>
  </body>
</html>
```

【ex7_1.jsp】

```
<%@ page contentType="text/html; charset=gb2312" language="java" %>
<html>
  <head>
    <title>选举结果</title>
  </head>
  <body>
    <center>
      <%!int sum[]=new int[3]; %>
      <%
          String num=request.getParameter("num");
          synchronized(sum)
          {
```

```
13                  if(num!=null)
14                  {
15                      if(num.compareTo("A")==0)
16                        sum[0]++;
17                      if(num.compareTo("B")==0)
18                        sum[1]++;
19                      if(num.compareTo("C")==0)
20                        sum[2]++;
21                  }
22              }
23          %>
24          <font color=#ffggkk size=6>选举结果<br /></font>
25          <hr color="#ffggkk" size="2" width="60%"></hr>
26          <font color=#996611 size=4>
27              候选人 A: <%=sum[0]%><br />
28              候选人 B: <%=sum[1]%><br />
29              候选人 C: <%=sum[2]%><br /><br />
30          </font>
31        </center>
32      </body>
33    </html>
```

【补充说明】

ex7.jsp 代码说明

第 7 行到第 19 行是一个 form 表单，设置了表单数据会传递给 ex7_1.jsp 代码页；第 13 行到 15 行放置了三个单选按钮，它们的标识分别为 A、B、C，名字都用 num；第 16 行放置了一个提交按钮，按钮上文字设置为投票，按钮的变量名为 tou。

ex7_1.jsp 代码说明

第 8 行声明了一个数组；第 9 行到第 23 行是程序片；第 10 行是接收 ex7.jsp 页面中表单传递过来的单选按钮的变量值；第 11 行到第 22 行是用 Synchronized 来定义数组 sum，这样就可以避免多个用户同时投票时产生的混乱，只要 Tomcat 服务器不重启，多次打开 IE 调用这个 JSP 页面，可以观察到 sum 数组可以完成连续统计投票结果；第 13 行到第 21 行是根据 num 的标识来判断给哪位候选人累计票数；第 27 到第 29 行是表达式输出，输出三个候选人的票数。

3.3 JSP 指令

实战演练 1——使用 JSP 指令显示当前时间

【题目】

用不同的输出方式显示当前时间。

【分析】

本题目主要是让大家体会 include 指令的用法。这个示例由两个页面来完成，使用

include 指令来读入能够显示当前的日期和时间的页面。

【运行效果】

打开开发工具 MyEclipse，在上一个题目所建立的工程 jsp-yufa 中的 WebRoot 文件夹下，新建工程文件 qj_time.jsp 和 dateshow.jsp。程序输入完毕并且保存之后，运行开发工具 MyEclipse 中的 Tomcat 服务器。最后在浏览器中输入地址 http://127.0.0.1:8080/jsp-yufa/qj_time.jsp 就可以看到运行结果了（如图 3-11 所示）。

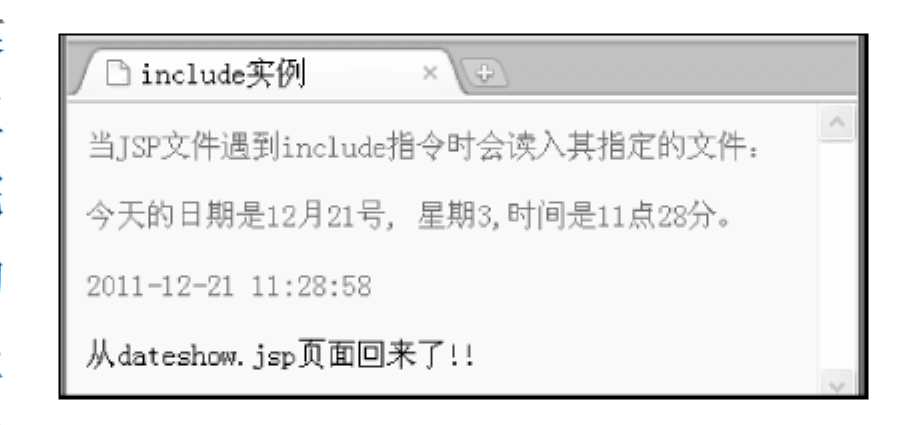


图 3-11 运行 qj_time.jsp 页面显示当前时间

【源代码】

【qj_time.jsp】

```
1  <%@page language="java" contentType="text/HTML;charset=gb2312" %>
2  <%@page import="java.util.*"%>
3  <HTML>
4    <head>
5      <title>include 实例</title>
6    </head>
7    <body bgcolor="#FFFFCC">
8      <font color="#946714" size="3"> 当 JSP 文件遇到 include 指令时会读入其指定的文件：
9        <%@ include file="dateshow.jsp" %>
10     </font>
11     <p>从 dateshow.jsp 页面回来了!!
12   </body>
13 </HTML>
```

【dateshow.jsp】

```
<%@page language="java" contentType="text/HTML;charset=gb2312"%>
<%@page import="java.util.*"%>
<%
  //变量 today 被声明为 Date 类型，可以得到时间信息
  Date today=new Date();
%>
<p>今天的日期是<%=today.getMonth()+1 %>月<%=today.getDate() %>号， 星期
<%=today.getDay() %>，时间是<%=today.getHours() %>点<%=today.getMinutes()
%>分。
<p><%=today.toLocaleString() %>
```

【补充说明】

qj_time.jsp 代码说明

第 2 行是使用 page 指令的 import 属性引入了 Java 的 util 工具包，在这个包中，Java 提供了一些实用的方法和数据结构。例如，Java 提供日期（Date）类、日历（Calendar）类来产生和获取日期及时间，因为我们这个题目中用到了 Data 类，所以要引入这个工具包。第 9 行是使用 include 指令读入 dateshow.jsp 页面的代码执行完 dateshow.jsp 页面的代

码之后会返回本页面继续执行后面的语句。插入点为 include 语句的位置。

dateshow. jsp 代码说明

第 2 行是使用 page 指令的 import 属性引入了 Java 的 util 工具包；第 5 行使用到了 Date 类，通过声明 Date 类的对象 today 来引用一些方法获取当前的日期和时间；第 7 行到第 8 行是使用 today 对象调用 Date 类的方法来实现时间日期的表达式输出。

在这个例题中，dateshow. jsp 页面属于被包含的文件，被包含的文件中不能含有<HTML>、</HTML>、<body>或</body>标签。因为被包含的文件的全部内容将被插入 JSP 文件中 include 指令所在的地方，这些标签将会同 JSP 文件中已有的同样的标签发生冲突。

3.3.1 page 指令

page 指令的作用范围是整个 JSP 文件和该 JSP 文件用 include 指令或<jsp:include>包含进来的任何静态文件，整个 JSP 文件和这些静态文件一起称为一个“平移单元”。page 指令不适用于任何动态的包含文件。可以在一个“平移单元”使用多个 page 指令，但是每一个属性只能使用一次，除了 import(因为 import 属性和 Java 程序语言的 import 语句很类似，能够多次使用，就像在 Java 语言中能够多次使用 import 命令一样)。不论用户将 page 指令放到 JSP 文件或被包含的文件的任何一个位置，它的作用范围都是整个“平移单元”。然而，一个好的编程风格是常常将它放到文件的顶部。

例如上面题目中用到的语句：

```
<%@page language="java" contentType="text/HTML;charset=gb2312" %>
<%@page import="java.util.*"%>
```

这其中用到的属性有以下三种。

1. language 属性

language="java"表示在 JSP 文件的脚本段、声明和表达式中使用的语言。

2. contentType 属性

该属性用来设置 JSP 页面的 MIME type 和 character encoding(字符编码)，用于响应客户端的请求。contentType="text/HTML;charset=gb2312"表示设置的 MIME type 是 text/HTML，字符编码集是 gb2312。

3. import 属性

该属性用来导入页面中要用到的包或类，导入的包或类可以是 Java 环境的核心类，也可以是用户自己编写的包或类。用法为：<%@page import="java.util.*"%> JSP 文件中可以导入一个或多个用逗号隔开的包或类，例如：

```
<%@ page import="java.io.*","java.util.*","java.sql.*" %>
```

默认情况下，JSP 页面 import 属性有以下值：

java.lang.*、javax.servlet.*、javax.servlet.jsp.*、javax.servlet.http.*。

3.3.2 include 指令

include 指令可以在 JSP 文件中插入一个包含文本和代码的文件。被包含的文件可以是 JSP 文件、HTML 文件或文本文件。

语法格式如下：

```
<%@include file="relativeURL"%>
```

上面例题中的用法如下：

```
<%@ include file="dateshow.jsp" %>
```

如果被包含的文件是 JSP 文件，则 JSP 引擎编译完该 JSP 文件后，执行的结果将插入到主 JSP 文件中 include 指令所在的位置；如果被包含的文件是 HTML 文件或文本文件，则 JSP 引擎不对其进行编译，直接将其内容插入主 JSP 文件中 include 指令所在的位置。

该包含是静态包含，即被包含的文件处理完，而且结果也插入主 JSP 文件完毕后，主 JSP 文件将继续执行 include 指令下面的内容。

注意：

(1) 被包含的文件中不能含有<html>、</html>、<body>或</body>标签。因为被包含的文件的全部内容将被插入 JSP 文件中 include 指令所在的地方，这些标签将会同 JSP 文件中已有的同样的标签发生冲突。

(2) 假如被包含的文件发生变化，主 JSP 页面将被重新编译。

3.3.3 Date 对象

Date 对象是一个有关日期和时间的对象。它具有动态性，即必须使用 new 运算符创建一个实例。例如：

```
Date today=new Date();
```

Date 对象没有提供直接访问的属性，只具有获取和设置日期和时间的属性。

实战演练 2——include 指令练习

【题目】

需要完成的功能是链接前面的任意三个例题，本题以链接 JSP 页面结构体验、表达式练习、方法声明练习为例。

【分析】

本题使用 include 指令调入 head.jsp、body.jsp、foot.jsp 三个文件来完成 ex6.jsp 的页面功能显示。

【运行效果】

打开开发工具 MyEclipse，在工程 jsp-yufa 中的 WebRoot 文件夹下，新建工程文件

ex6.jsp、head.jsp、body.jsp、foot.jsp。程序输入完毕并且保存之后，运行开发工具 MyEclipse 中的 Tomcat 服务器。在浏览器中输入地址 http://127.0.0.1:8080/jsp-yufa/ex6.jsp 就可以看到运行结果了（如图 3-12 所示）。



图 3-12 ex6.jsp

【源代码】

【ex6.jsp】

```
<%@page language="java" contentType="text/html;charset=gb2312" import="java.util.*"%>
<html>
  <center>
    <%@ include file="head.jsp"%>
    <%@ include file="body.jsp"%>
    <%@ include file="foot.jsp" %>
  </center>
</html>
```

【head.jsp】

```
<%@page language="java" contentType="text/html;charset=gb2312" import="java.util.*"%>
<font color=#00ccaa size="6">include 指令实例</font>
```

【body.jsp】

```
<%@page contentType="text/html;charset=gb2312" import="java.util.*"%>
<hr size="2" width="60%" color="#00ccaa">
<p><a href="ex4.jsp">1、JSP 页面结构体验</a>
<p><a href="ex2.jsp">2、表达式练习</a>
<p><a href="ex5.jsp">3、方法声明练习</a>
<hr size="2" width="60%" color="#00ccaa">
```

【foot.jsp】

```
<%@page contentType="text/html;charset=gb2312" import="java.util.*"%>
<%
  //变量 today 被声明为 Date 类型,可以得到时间信息
  Date today=new Date();
%>
<p>今天的日期是:
<%=today.toLocaleString() %>
```

【补充说明】

ex6.jsp 代码说明

第 1 行是 page 指令,其中 import 属性引入了 util 工具包,因为后面会用到日期类;第 4 行到第 6 行都是 include 指令的语句,分别引入了 head.jsp、body.jsp、foot.jsp 三个文件来完成 ex6.jsp 的页面功能显示。

head.jsp 代码说明

第 2 行代码用于设定指定文字的格式。

body.jsp 代码说明

第 2 行和第 6 行使页面插入了彩色的水平线;第 3 行到第 5 行设置了三个链接。

foot.jsp 代码说明

第 4 行新建了一个 Date 类的对象 today;第 7 行用于把当前时间以表达式的形式输出。

3.4 taglib 指令

实战演练——taglib 指令引入页脚的练习

【题目】

使用 taglib 指令完成版权信息的引入。

【分析】

本题目介绍了 taglib 指令的用法,首先运行 a.jsp 页面,在页面上显示了一个表格,内容为本章一些题目的链接,在页脚处引用 a.tag 文件显示一些版权信息。

【运行效果】

打开开发工具 MyEclipse,在上一个题目所建立的工程 jsp-yufa 中的 WebRoot 文件夹下,新建工程文件 a.jsp,在 WEB-INF 文件夹下新建文件夹 tags,在 tags 文件夹中新建 a.tag 文件(如图 3-13 和图 3-14 所示)。程序输入完毕并且保存之后,运行开发工具 MyEclipse 中的 Tomcat 服务器。最后在浏览器中输入地址 http://127.0.0.1:8080/jsp-yufa/a.jsp 就可以看到运行结果了(如图 3-15 所示)。

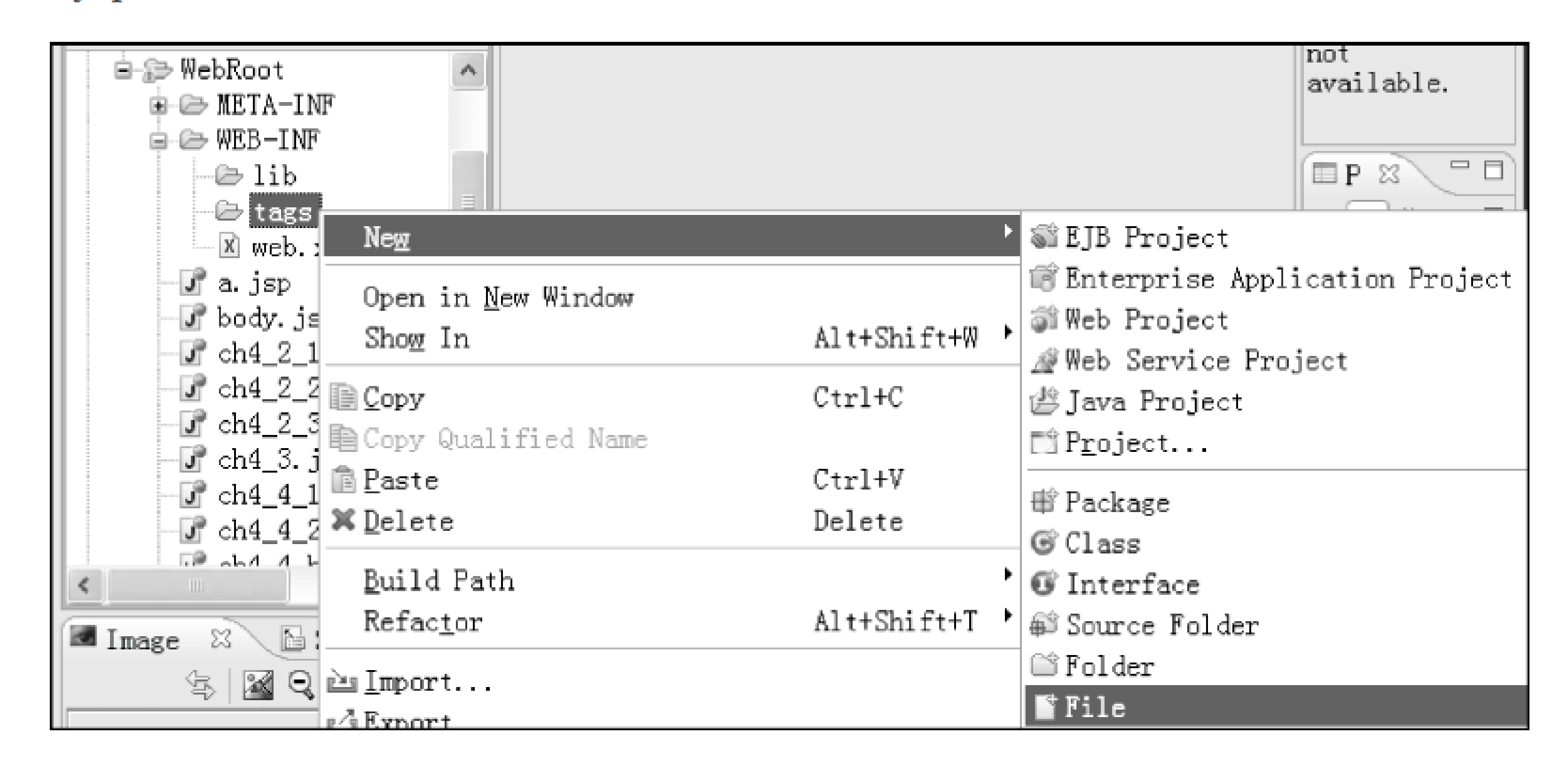


图 3-13　新建 tags 文件夹

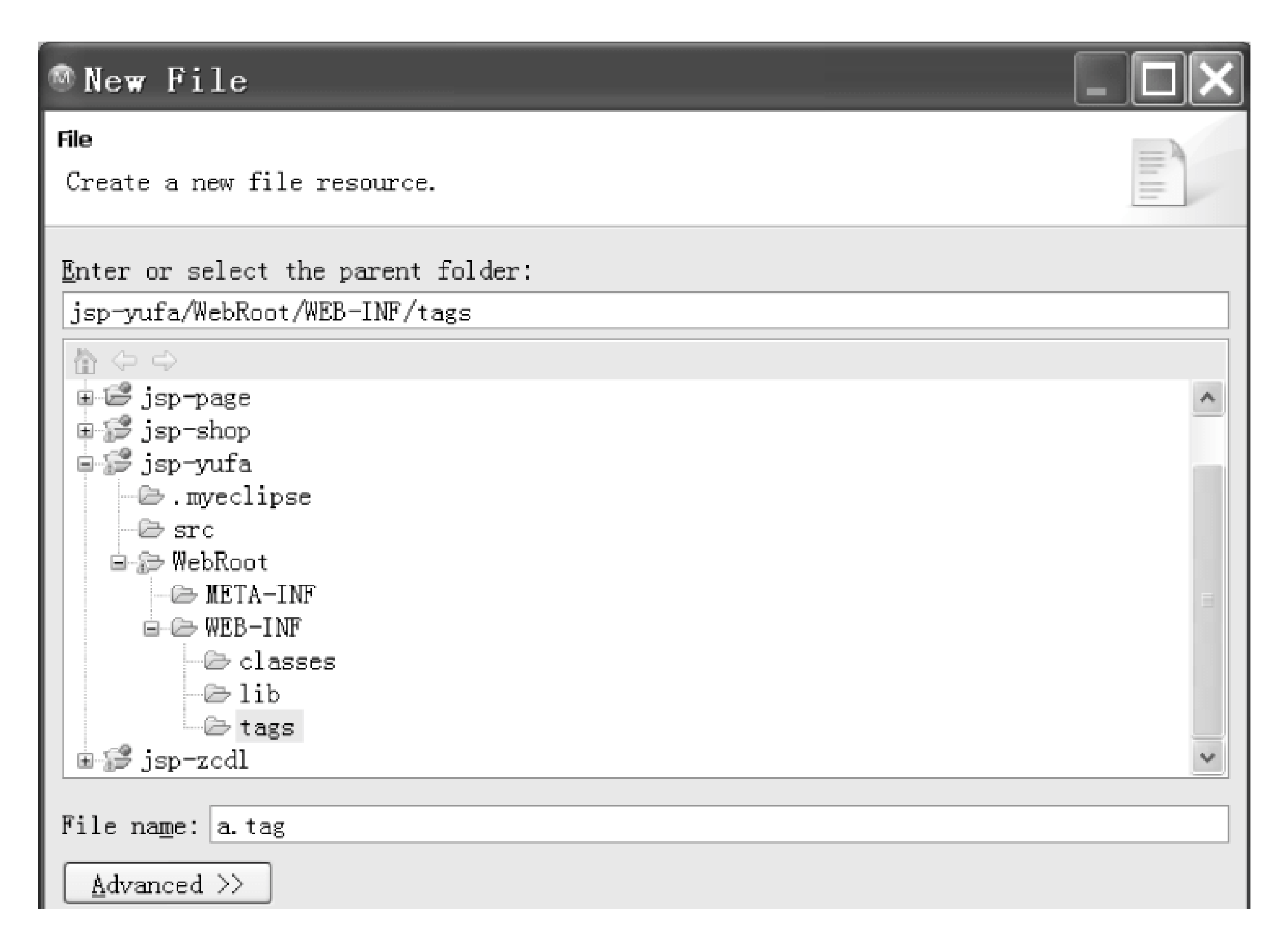


图 3-14　新建 a. tag 文件



图 3-15　a. jsp 页面

【源代码】

【a. tag】

```
<%@ tag pageEncoding="UTF-8"%>
<center>
  <p><hr width="95%" size="2" color="#AA4E11"></hr></p>
  <table width="90%" align="center">
    <tr><td align="center">本页面使用 taglib 指令</td></tr>
    <tr><td align="center">Copyright★2011 E-mail:sunqian_teacher@163.com</td></tr>
  </table>
</center>
```

【a. jsp】

```
<%@ page contentType="text/HTML; charset=gb2312" language="java" %>
<%@ taglib prefix="tags" tagdir="/WEB-INF/tags"%>
<HTML>
```

```
4    <head>
5      <title>taglib 范例</title>
6    </head>
7    <body>
8      <table width="400" border="1" align="center" cellpadding="8" cellspacing="0"
       bordercolor="#999999">
9        <tr>
10         <td align="center" rowspan="1" colspan="2">情境 3---JSP 语法</td>
11       </tr>
12       <tr>
13         <td align="center">题目 1</td>
14         <td><a href="ex1_1.jsp">显示当前时间</a></td>
15       </tr>
16       <tr>
17         <td align="center">题目 2</td>
18         <td><a href="qj_time.jsp">计算三角形面积</a></td>
19       </tr>
20       <tr>
21         <td align="center">题目 3</td>
22         <td><a href="ex3.HTML">登录</a></td>
23       </tr>
24     </table>
25     <tags:a/>
26   </body>
27 </HTML>
```

【补充说明】

a. tag 代码说明

代码中书写的内容为一些版权信息，相当于自己定制的一个标签。

a. jsp 代码说明

第 2 行是使用 taglib 指令引入标签，即 a.tag 代码页的内容，其中标签的前缀为 tags，标签路径为/WEB-INF/tags；第 25 行表示在这个位置插入标签，前缀名为 tags，标记名为 a。

taglib 指令

taglib 指令用于定义一个标签库及其自定义标签的前缀。taglib 指令用于声明此 JSP 文件中使用了自定义的标签，同时引入标签库，并指定它们的前缀。

标记文件的扩展名为.tag，它与 JSP 文件一样可以引入其他的标签文件的内容。此外，标签文件必须放在网站根目录下的/WEB-INF/tags/目录或其子目录下，如果标签文件被包装成 JAR 文件，则需要放在/WEB-INF/lib/目录下。

语法格式如上面例题所用：

```
<%@ taglib prefix="tags" tagdir="/WEB-INF/tags"%>
```

其中 prefix 属性定义的字符串为标记前缀，在 JSP 页面中使用自定义标记时，一般格式为<前缀：标记名>，标记名为标记库中自定义标记的具体名称。上面题目中前缀为 tags，标记名为 a。

3.5 JSP 动作 include、param、forward

实战演练 1——登录

【题目】

本题要完成学生和教师的分别登录功能。

【分析】

题目用 5 个页面完成，分别是 ex3.html、ex3.jsp、ex3_1.jsp、ex3_2.jsp、return.jsp。我们设计 ex3.html 网页是分权限登录界面，在这个界面需要设计一个表单，表单中有一个表格，在表格中有两个文本框用于输入用户名和密码；有两个单选按钮用于选择不同的权限；一个重置按钮用于重新输入信息；一个提交按钮用于提交界面的信息到 ex3.jsp 代码页。ex3.jsp 代码页是没有界面设计的，在此代码页中需要使用 request 内置对象的 get 方法接收 ex3.html 页面提交的登录信息，然后通过一个 if 条件语句来判断不同的权限，使用 forward 动作将不同权限定位到不同的二级页面，并且使用 param 动作配合 forward 动作使登录信息能够继续传递到不同权限的二级页面中。网页 ex3_1.jsp 是学生权限登录的二级页面，ex3_2.jsp 是教师权限登录的二级页面，代码页中使用 include 动作来引入 return.jsp 代码页，为的是在二级页面中能够显示登录人员的登录信息。return.jsp 代码页使用 request 内置对象的 get 方法接收登录人员的登录信息，并且使用表达式的方法输出登录信息。return.jsp 代码页中的代码是 ex3_1.jsp 页面和 ex3_2.jsp 页面能够公用的代码，所以把这部分代码使用一个代码页单独写出来，然后在应用的时候使用 include 动作来引用，这样就不用分别在 ex3_1.jsp 页面和 ex3_2.jsp 页面中写入重复的代码了。

【运行效果】

打开开发工具 MyEclipse，在上一个题目所建立的工程 jsp-yufa 中的 WebRoot 文件夹下，新建工程文件 ex3.html、ex3.jsp、ex3_1.jsp、ex3_2.jsp、return.jsp。程序输入完毕并且保存之后，运行开发工具 MyEclipse 中的 Tomcat 服务器。最后在浏览器中输入地址 http://127.0.0.1:8080/jsp-yufa/ex3.html 就可以看到运行结果了(如图 3-16～图 3-18 所示)。



图 3-16 ex3.html 登录界面

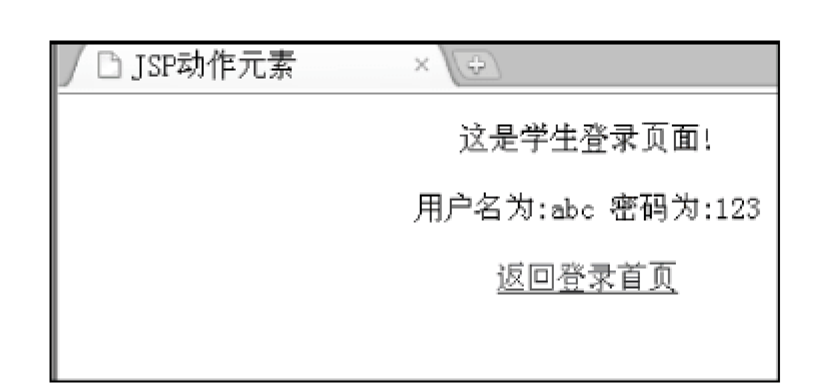


图 3-17 ex3_1.jsp 学生界面

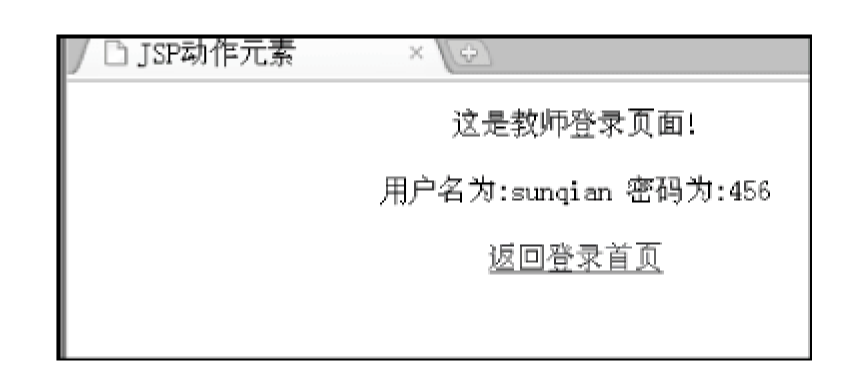


图 3-18 ex3_2.jsp 教师界面

【源代码】

【ex3. html】

```
<HTML>
  <head>
    <title>JSP 动作元素</title>
  </head>
  <body>
    <form method="post" action="ex3.jsp">
      <center>
        <table width="500" border="0" cellspacing="0" cellpadding="10" bgcolor="#FFFFCC">
          <tr>
            <td width="250" align="center">用户名:</td>
            <td width="263"><input type="text" size="20" name="username" /></td>
          </tr>
          <tr>
            <td align="center">密   码: </td>
            <td><input type="password" size="20" name="password" /></td>
          </tr>
          <tr>
            <td align="center">
            <input type="radio" checked="checked" value="student" name="select" />学 生</td>
            <td>
            <input type="radio"  value="teacher" name="select" />教 师</td>
          </tr>
          <tr>
            <td align="center">
            <input type="submit" name="login" value="提交"/></td>
            <td><input type="reset" name="reset" value="重写"/></td>
          </tr>
        </table>
      </center>
    </form>
  </body>
</HTML>
```

【ex3. jsp】

```
<%@ page contentType="text/HTML; charset=gb2312" language="java" %>
<HTML>
  <head>
    <title>动作语法</title>
  </head>
  <body>
    <%
        String user,pwd,choice;
        user=request.getParameter("username");
        pwd=request.getParameter("password");
```

```
            choice=request.getParameter("select");
            if(choice.equals("student")){
        %>
        <!--如果是学生登录则将页面定向到 ex3_1.jsp-->
          <jsp:forward page="ex3_1.jsp">
          <jsp:param name="username" value="<%=user%>"/>
          <jsp:param name="password" value="<%=pwd%>"/>
          </jsp:forward>
        <%
          }else{
        %>
        <!--如果是教师登录则将页面定向到 ex3_2.jsp-->
          <jsp:forward page="ex3_2.jsp">
          <jsp:param name="username" value="<%=user%>"/>
          <jsp:param name="password" value="<%=pwd%>"/>
          </jsp:forward>
        <%
          }
        %>
    </body>
</HTML>
```

【ex3_1.jsp】

```
<%@ page contentType="text/HTML; charset=gb2312" language="java" %>
<HTML>
  <head>
    <title>动作语法</title>
  </head>
  <body>
    <center>
        这是学生登录页面!
        <P><jsp:include page="return.jsp"></jsp:include> </P>
    </center>
  </body>
</HTML>
```

【ex3_2.jsp】

```
<%@ page contentType="text/HTML; charset=gb2312" language="java" %>
<HTML>
  <head>
    <title>动作语法</title>
  </head>
  <body>
    <center>
        这是教师登录页面!
        <P><jsp:include page="return.jsp"></jsp:include> </P>
    </center>
  </body>
```

```
12 </HTML>
```

【return.jsp】

```
1 <%@ page contentType="text/HTML; charset=gb2312" language="java" %>
2 <HTML>
3   <head>
4     <title>动作语法</title>
5   </head>
6   <body>
7     <center>
8       <%
9           String user,pwd;
10          user=request.getParameter("username");
11          pwd=request.getParameter("password");
12      %>
13      <p>用户名为:<%=user%>密码为:<%=pwd%> </p>
14      <P><a href="ex3.HTML">返回登录首页</a></P>
15    </center>
16  </body>
17 </HTML>
```

【补充说明】

ex3.html 代码说明

第 6 行到第 30 行是一个完整的表单,定义表单的数据最终提交到 ex3.jsp 页面执行;第 8 行到第 28 行是表单中的表格,表格分为 4 行 2 列,每一组<tr></tr>就表示表格的一行;第 11 行是表格的第 1 行第 2 列,在这个位置放置一个文本框,用于输入用户名,变量名为 username;第 15 行是表格的第 2 行第 2 列,在这个位置放置一个文本框,用于输入密码,变量名为 password;第 19 行和第 21 行是表格的第 3 行第 1 列和第 2 列,在这两个位置放置两个单选按钮,初始值分别设置为 student 和 teacher,变量名都设置为 select;第 25 行和第 26 行是表格的第 4 行第 1 列和第 2 列,在这两个位置放置两个按钮,一个用于提交表单的数据,一个用于重新填写登录信息。

ex3.jsp 代码说明

第 7 行到第 13 行是程序片,其中第 9 行到第 11 行是接收 ex3.html 页面表单传递过来的用户名、密码、角色选择;第 12 行是一个条件语句,用来根据不同的角色执行不同的任务;第 15 行到第 18 行是使用 forward 动作对页面进行重新定向,并且使用 param 动作把本页面的参数传递到 forward 动作指定的页面中;第 23 行到第 26 行也是使用 forward 动作和 param 动作配合来完成参数的传递和页面的重新定向。

ex3_1.jsp 和 ex3_2.jsp 代码说明

第 9 行使用 include 动作,引入了 return.jsp 页面。

return.jsp 代码说明

第 10 行和第 11 行接收用户名和密码的参数,这两个参数是 ex3.jsp 代码页中 param 动作传递到 ex3_1.jsp 和 ex3_2.jsp 页面中的参数,因为 return.jsp 代码页是被 ex3_1.jsp 和 ex3_2.jsp 文件引入的,所以可以接收这两个参数。

3.5.1 <jsp:include> 动作标记

<jsp:include>动作标记用于向当前的页面中包含其他的文件，这个文件可以是动态文件也可以是静态文件。该标识的使用格式如下：

```
<jsp:include page="被包含文件的路径" />
```

如上面例题中用到的：

```
<jsp:include page="return.jsp"></jsp:include>
```

该标记也用于向被包含的动态页面中传递参数：

```
<jsp:include page="被包含文件的路径" >
<jsp:param name="参数名称" valude="参数值"/>
</jsp:include>
```

如下面例题中用到的：

```
<jsp:include page="ex8_1.jsp" >
<jsp:param name="a" value="<%=a%>" />
<jsp:param name="b" value="<%=b%>" />
</jsp:include>
```

(1) Page 属性指定了被包含文件的路径，其值可以是一个代表了相对路径的表达式。当路径是以“/”开头时，则按照当前应用的路径查找这个文件；如果路径是以文件名或目录名称开头，那么将按照当前的路径来查找被包含的文件。

(2) <jsp:param>子标识可以向被包含的动态页面传递参数。

3.5.2 <jsp:include> 动作标记与 include 指令标记的区别

如果使用<jsp:include>动作标记，被包含文件的语法和逻辑独立于当前页面，单独被编译，当前页面执行时再单独处理包含文件的速度较慢，但是可以与 param 标记组合使用，这样更加灵活，如果遇到需要传递参数的情况更适合使用<jsp:include>动作标记。

如果使用 include 指令，被包含文件在 include 指令的地方插入，与当前页面合并成一个文件，一同执行，执行速度比<jsp:include>动作标记要快。所以如果不需要传递参数的情况下，include 指令更适于使用。

3.5.3 param 动作

<jsp:param>动作标记不能单独使用，必须作为 jsp:include、jsp:forward、jsp:plugin 标记的子标记使用，并为它们提供参数。

语法格式如下：

```
<jsp:param name="参数名字" value="指定的参数值"/>
```

例如：

```
<jsp:forward page="ex3_2.jsp">
<jsp:param name="username" value="<%=user%>"/>
<jsp:param name="password" value="<%=pwd%>"/>
</jsp:forward>
```

当该标记作为 jsp:include 的子标记时，name 和 value 为页面之间提供参数传递。

3.5.4 forward动作

<jsp:forward>动作将客户端发来的请求重定向到另一个 JSP、Servlet 或 HTML 文件中。语法格式如下：

```
<jsp:forward page="要转向的页面 URL| <%=表达式%>" />
```

或者

```
<jsp:forward page="要转向的页面 URL | <%=表达式%>" >
<jsp:param name="属性名" value="属性值| <%=表达式%>" />
</jsp:forward>
```

例如：

```
<jsp:forward page="ex3_2.jsp">
<jsp:param name="username" value="<%=user%>"/>
<jsp:param name="password" value="<%=pwd%>"/>
</jsp:forward>
```

实战演练 2——比较大小

【题目】

在页面中显示文件什么时候被加载，什么时候加载完毕，并且完成两个参数的传递和比较大小。

【分析】

本题目需要新建两个文件 ex8.jsp、ex8_1.jsp 来完成所有的操作，在 ex8.jsp 中主要完成的是使用 include 和 param 动作完成文件的加载和参数值的传递；在 ex8_1.jsp 中主要完成的是参数值的接收和声明一个能够完成比较大小功能的方法。

【运行效果】

打开开发工具 MyEclipse，在上一个题目所建立的工程 jsp-yufa 中的 WebRoot 文件夹下，新建工程文件 ex8.jsp、ex8_1.jsp。程序输入完毕并且保存之后，运行开发工具 MyEclipse 中的 Tomcat 服务器。最后在浏览器中输入地址 http://127.0.0.1:8080/jsp-yufa/ex8.jsp 就可以看到运行结果了(如图 3-19 所示)。

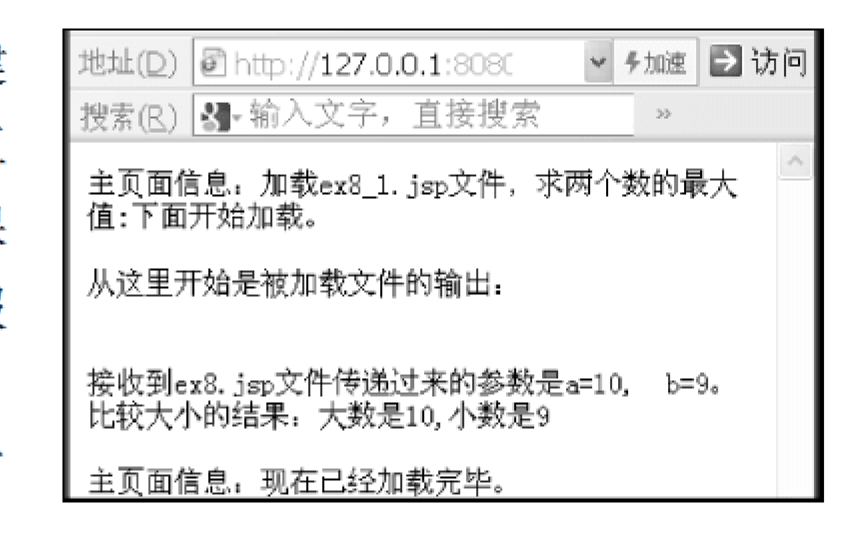


图 3-19　ex8.jsp 页面

【源代码】

【ex8.jsp】

```
<%@ page contentType="text/html; charset=gb2312" language="java" %>
<html>
  <head>
    <title>include 和 param 结合使用</title>
  </head>
  <body>
    <% int a=10,b=9;%>
    <p>主页面信息：加载 ex8_1.jsp 文件,求两个数的最大值:下面开始加载。
      <jsp:include page="ex8_1.jsp" >
        <jsp:param name="a" value="<%=a%>" />
        <jsp:param name="b" value="<%=b%>" />
      </jsp:include>
    </p>
    <p>主页面信息：现在已经加载完毕。
    </p>
  </body>
</html>
```

【ex8_1.jsp】

```
<%@ page contentType="text/html; charset=gb2312" language="java" %>
<html>
  <head>
    <title></title>
  </head>
  <body>
    <%! public String bijiao(int x,int y){
        if (x>y){
          int temp;
          temp=x;
          x=y;
          y=temp;
      }
      return ("大数是"+y+",小数是"+x);
    }
    %>
    <%
      int x=Integer.parseInt(request.getParameter("a"));
      int y=Integer.parseInt(request.getParameter("b"));
    %>
    <p>从这里开始是被加载文件的输出：</p><br/>
    接收到 ex8.jsp 文件传递过来的参数是 a=<%=x%>,  b=<%=y%>。<br/>
    比较大小的结果：<%=bijiao(x,y)%>
```

```
24    </body>
25 </html>
```

【补充说明】

ex8.jsp 代码说明

第 9 行是使用 include 动作加载 ex8_1.jsp 文件；第 10 行到第 11 行是使用 param 动作给两个变量赋值并且传递到被加载的文件 ex8_1.jsp；第 12 行表示 include 动作结束。

ex8_1.jsp 代码说明

第 7 行到第 16 行是一段方法声明，这个方法用来比较 ex8.jsp 文件传递过来的变量值的大小，并且有结果返回值；第 17 行到第 20 行是接收 ex8.jsp 文件传递过来的变量值，而且还为这两个变量做了整形的强制转换；第 22 行是输出本页面接收到的两个变量值；第 23 行是调用本页面声明的方法来对两个变量的大小进行比较，得出结果并输出。

3.6 useBean 动 作

JSP 中的 useBean 动作就是使用 Bean。Bean 是 JavaBean 的简称，实际上它就是 Java 开发中的一个类，通过封装属性和方法成为具有某种功能和接口的类。更简化一些来说，Bean 就是一个类，useBean 动作就是“使用类”的动作。

JSP+JavaBean 是 JSP 开发的一种典型模式，Sun 的 JavaBean 技术为 JSP 开发提供了极大的方便，JavaBean 是一个可重复使用的软件组件，使得页面与数据处理实现了真正的分离。

实战演练 1——应用 JavaBean 实现图书库存的录入并显示

【题目】

使用 useBean 动作设计一个图书的入库输入并显示的操作平台(不使用数据库)。

【分析】

要使用 useBean 动作来完成此题，首先应该创建一个图书的 Bean。在这个 Bean 中要包括图书名称和库存的 get 和 set 方法。本题需要两个界面来完成操作，一个界面用于录入；另一个界面用于显示。还有就是要解决图书录入时的汉字识别问题。所以综合下来本题总共需要四个文件，cart.java(图书的 Bean)、kucun.html(录入界面)、kucun_bean.jsp(显示结果页面)、hzsb.jsp(用于汉字识别的代码)。

【运行效果】

首先创建 Bean，打开开发工具 MyEclipse，在工程 jsp-yufa 的 src 文件夹中新建一个用于放置 Bean 的文件夹 bean(如图 3-20 和图 3-21 所示)。然后在 bean 文件夹中新建类 cart(如图 3-22 和图 3-23 所示)。在新建好的 cart.java 文件中输入程序源代码然后保存。当程序运行之后，cart.java 文件会自动转换成 cart.class 以供程序运作使用。

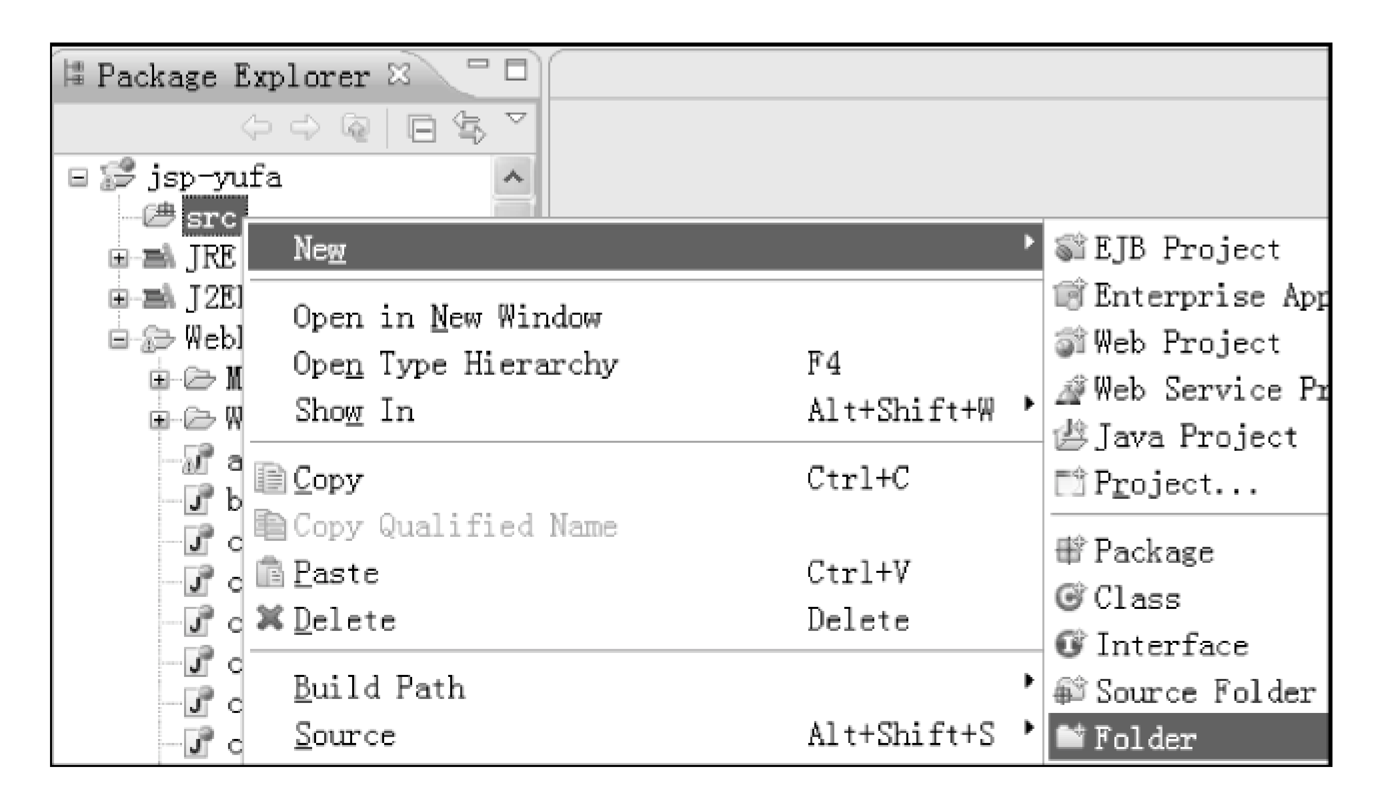


图 3-20 新建 JavaBean 的包

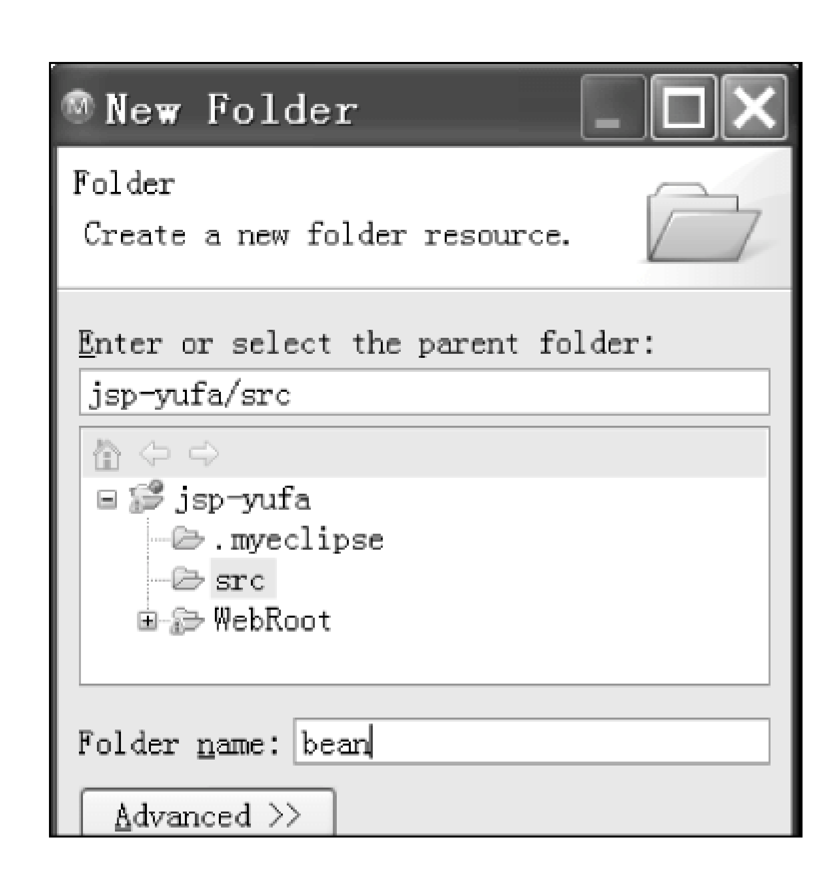


图 3-21 新建包名称为 bean

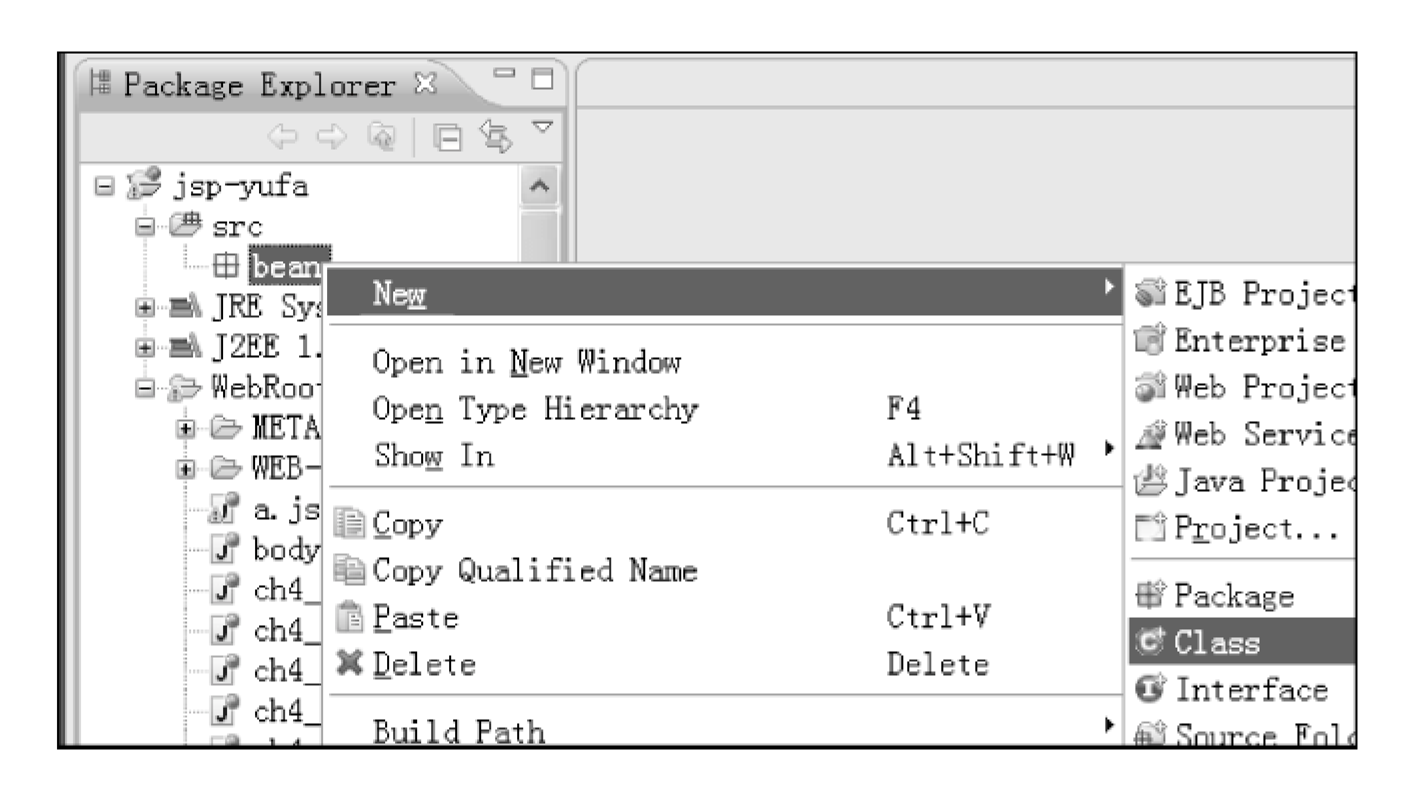


图 3-22 在 bean 包中新建类

在工程 jsp-yufa 中的 WebRoot 文件夹下，新建工程文件 kucun. html 和 kucun_bean. jsp。程序输入完毕并且保存之后，运行开发工具 MyEclipse 中的 Tomcat 服务器。最后在浏览器中输入地址 http://127. 0. 0. 1:8080/jsp-yufa/ kucun. html 就可以看到运行结果了(如图 3-24～图 3-26 所示)。

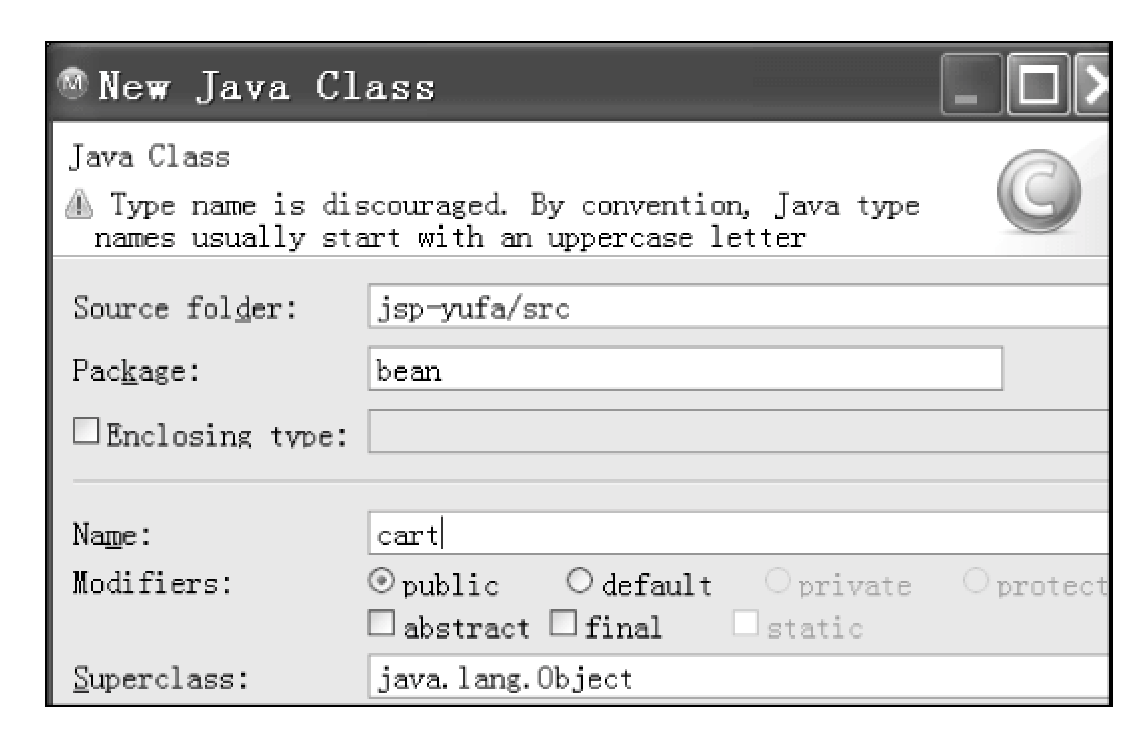


图 3-23　类名为 cart



图 3-24　kucun. html

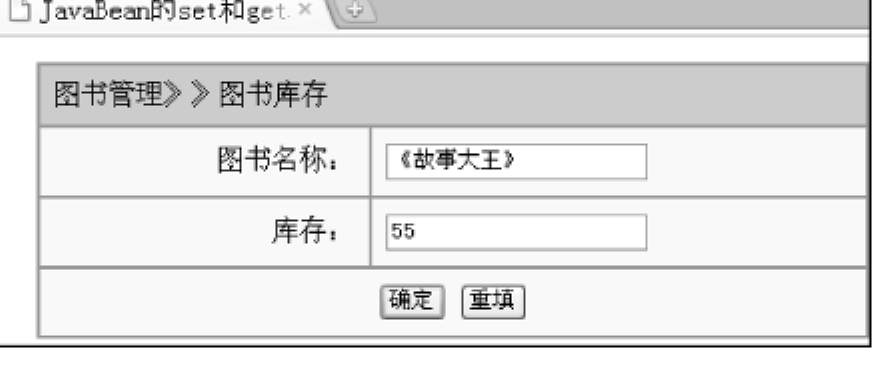



图 3-25　在 kucun. html 页面输入图书库存信息



图 3-26　kucun_bean. jsp

【源代码】

【cart. java】

```
package bean;
public class cart {
    private String bookname;
    private int num;
    public cart(){
        bookname="";
        num=0;
    }
    public cart(String bookname,int num){
        this.bookname=bookname;
        this.num=num;
    }
    public String getbookname(){
        return(bookname);
    }
    public void setbookname(String bookname){
        this.bookname=bookname;
    }
    public int getnum(){
        return(num);
    }
```

```
    public void setnum(int num){
        this.num=num;
    }
}
```

【kucun.html】

```
<html>
  <head>
     <title>JavaBean 的 set 和 get 示例</title>
  </head>
  <body>
    <form method="post" action="kucun_bean.jsp" name="form1">
      <table width="90%" border="1" align="center" cellpadding="8" cellspacing="0"
      bordercolor="#CC9900" bgcolor="#FFFFCC">
        <tr>
          <td height="30" colspan="2" align="left" valign="middle"
          bgcolor="#CCCCCC">图书管理》》图书库存</td>
        </tr>
        <tr>
          <td  align="right" valign="middle" width="40%">图书名称：</td>
          <td  align="left" valign="middle">
             <input type="text" name="bookname" id="textfield2"/></td>
        </tr>
        <tr>
           <td  align="right" valign="middle" >库存：</td>
           <td  align="left" valign="middle">
              <input type="text" name="num" id="textfield3"/></td>
        </tr>
        <tr>
           <td  colspan="2" align="center" valign="middle">
              <input type="submit" name="button" id="button" value="确定" />
              <input type="reset" name="button2" id="button2" value="重填" /></td>
        </tr>
      </table>
    </form>
  </body>
</html>
```

【hzsb.jsp】

```
<%@ page contentType="text/html;charset=gb2312" language="java" %>
<%!
    public String hzsb(String s)
    {
        String str=s;
        try
        {
```

```
                byte b[]=str.getBytes("ISO-8859-1");
                str=new String(b);
                return str;
            }
            catch(Exception e)
            {return str;}
        }
%>
```

【kucun_bean.jsp】

```
<%@ page contentType="text/html;charset=gb2312" language="java" %>
<%@ include file="hzsb.jsp" %>
<html>
  <head>
    <title>JavaBean 的 set 和 get 示例</title>
  </head>
  <body>
    <%
      String bookname=hzsb(request.getParameter("bookname"));
      int num=Integer.parseInt(request.getParameter("num"));
    %>
    <jsp:useBean id="book" class="bean.cart" scope="page" />
    <jsp:setProperty name="book" property="bookname" value="<%= bookname%>"/>
    <jsp:setProperty name="book" property="num" value="<%= num%>"/>
    <table width="90%" border="1" align="center" cellpadding="10" cellspacing="0" bordercolor="#CC9900" bgcolor="#FFFFCC">
      <tr>
          <td colspan="2" align="left" valign="middle" bgcolor="#CCCCCC">使用javabean 技术的 setProperty 属性设置参数值,getProperty 属性获取参数值:</td>
      </tr>
      <tr>
        <td align="center" valign="middle" width="40%">图书名称:</td>
        <td align="left" valign="middle">
          <jsp:getProperty name="book" property="bookname"/><br></td>
      </tr>
      <tr>
        <td align="center" valign="middle" >    库存:</td>
        <td align="left" valign="middle">
          <jsp:getProperty name="book" property="num"/><br></td>
      </tr>
    </table>
    <p align="center"><a href="js.jsp">返回到访问计数器页面</a>
  </body>
</html>
```

【补充说明】

cart.java 代码说明

第 1 行表示 cart.java 在 bean 这个文件夹中；第 2 行到第 25 行是一个完整的 cart 类的声明，这里大家应该已经发现 cart 类是一个公共类，类的名字是 MyEclipse 工具自动生成的并且与文件名相同，因为在程序运行的过程中，cart.java 将转变成 cart.class 提供给程序使用，所以公共类的名字与文件名相同非常重要。第 3 行到第 4 行是 cart 类中的两个私有变量，bookname 表示图书名称，num 代表图书库存；第 5 行到第 8 行是 cart 类的构造函数，是不带参数的，用于对 cart 的对象进行初始化；第 9 到第 12 行也是 cart 类的构造函数，是带参数的，用于对 cart 的对象进行初始化；第 13 行到第 15 行是获取图书名称的方法，可以返回图书名称的字符串；第 16 行到第 18 行是设置图书名称；第 19 行到第 21 行是获取图书库存，可以返回整形值；第 22 行到第 24 行是设置图书库存。

kucun.html 代码说明

这个页面是做的界面设计，第 6 行到第 27 行是一个完整的表单，表单中的数据将要提交到 kucun_bean.jsp 代码页；第 7 行到第 26 行是一个表格，表格总共有 4 行 2 列；代码第 8 行到第 10 行是表格的第一行，这行把两个单元格合并为 1 个，并且设置了该行的格式和颜色等属性；代码第 11 行到第 15 行是表格的第二行，这行的第二个单元格放置了一个文本框，用于接收输入图书名称的字符串，变量名为 bookname；第 16 行到第 20 行是表格的第三行，这行的第二个单元格放置了一个文本框，用于接收输入库存的字符串，变量名为 num；第 21 行到第 25 行是表格的第四行，这行的两个单元格合并为 1 个，放置了两个按钮一个用于提交表单的数据，另外一个用于表格内容的重新输入。

hzsb.jsp 代码说明

这是一段对汉字进行解码的程序，request 对象在获取提交的汉字字符时，会出现乱码，解决方法就是将字符串用 ISO-8859-1 进行编码，并将编码放到一个字节数组中，然后再用这个字节数组创建 String 对象即可。

kucun_bean.jsp 代码说明

第 2 行表示引入 hzsb.jsp 文件；第 8 行到第 11 行接收 kucun.html 表单提交的图书名称和库存的数据，并且把图书名称做了汉字识别转换，把库存做了强制整形转换；第 12 行表示使用 useBean 动作，在此动作中用到的类是 bean 包下的 cart 类，建立类的对象是 book，使用范围是 page；第 13 行表示设置对象 book 的 bookname 属性值为 kucun.html 表单传递过来的图书名称；第 14 行表示设置对象 book 的 num 属性值为 kucun.html 表单传递过来的库存量；第 15 行到第 29 行是一个完整的表格，总共有 3 行 2 列；第 16 行到第 18 行是表格的第一行，在这行里把两个单元格做了合并，并且设置了单元格的对齐方式和颜色；第 19 行到第 23 行是表格的第二行，在这行显示图书的名称，其中第二个单元格也就是代码第 22 行是获取对象 book 的 bookname 属性值；第 24 行到第28 行是表格的第三行，在这行显示图书的库存，其中第二个单元格也就是代码第 27 行是获取对象 book 的 num 属性值；第 30 行是一个其他页面的链接，与后面的习题相关联。

3.6.1　创建 Bean

Bean 分为可视化组件和非可视化组件，非可视化的 Bean 被广泛地使用在 JSP 编程中。上面的例题就是属于非可视化的组件。非可视化 Bean 的一些规定如下。

1. Bean 的构造方法

Bean 就是符合一定条件的 Java 类，该类必须声明为 public 类型，可供其他类实例化。类中如果有构造方法，必须声明为 public 类型。构造方法也就是我们常说的构造函数，构造方法是一种特殊的方法，具有以下特点。

（1）构造方法的方法名必须与公共类名相同。

（2）构造方法没有返回类型，也不能定义为 void，在方法名前面不声明方法类型。

（3）构造方法的主要作用是完成对象的初始化工作，所以构造方法可以没有参数也可以有多个参数，它能够把定义对象时的参数传给对象的域。

（4）构造方法总是伴随着 new 操作一起调用。

（5）一个类可以定义多个构造方法，如果在定义类时没有定义构造方法，则编译系统会自动插入一个无参数的默认构造器，这个构造器不执行任何代码。

2. Bean 的属性与方法

Bean 的属性必须声明为 private，方法必须声明为 public 类型。Bean 中用一组 set 方法设置 Bean 的私有属性值，get 方法获得 Bean 的私有属性值。set 和 get 方法名称与属性名称之间必须对应，也就是：如果属性名称为 xxx，那么 set 和 get 方法的名称必须为 setxxx()和 getxxx()，set 和 get 为前缀与属性名字首字母大写组成 set 和 get 方法名称。另外，创建 Bean 必须带有包名。

3.6.2　<jsp:useBean> 动作标记

1. <jsp:useBean>动作标记

要想在 JSP 页面中使用 Bean，必须首先使用<jsp:useBean>动作标记在页面中定义一个 Bean 的实例，这个被定义实例有一定的生存范围及一个唯一的 id，JSP 页面通过 id 来识别 Bean，也可以在程序片中通过 id. method 形式来调用 Bean 中的 set 和 get 方法。下面是 useBean 动作标记的使用格式：

```
<jsp:useBean id="bean 名字" scope="page|request|session|application" class="packageName.
calssName"/>
```

上面题目中的用法：

```
<jsp:useBean id="book" class="bean.cart" scope="page" />
```

这句话中，使用到的类是 bean 包下的 cart 类，book 是类的对象，使用范围是 page。

2. 使用范围分类

(1) Bean 的作用范围是 page：同一个客户访问的每个页面的 Bean 对象是互不相同的，它们占用不同的存储空间，当客户离开这个页面时，释放占用的内存空间。不同客户访问同一页面时，Bean 之间互不干扰。

(2) Bean 的作用范围是 session：同一个客户在访问同一个站点的不同 JSP 页面时，只要 Bean 的 id 相同，范围是 session，该客户使用的就是同一个 Bean，无论在哪个页面对 Bean 的属性进行了改变，其他页面都能看到。不同客户之间的 Bean 是不同的、互不影响的。

(3) Bean 的作用范围是 request：同一个客户在 request 作用范围内，请求与相应之间共用一个 id 的相同的 Bean，请求响应完成时 Bean 即被释放，不同客户的 Bean 互不相同。

(4) Bean 的作用范围是 application：同一站点下的所有用户共享一个 Bean，任何客户对 Bean 的修改都会影响其他客户。

3.6.3 <jsp:setProperty> 动作标记

使用<jsp:setProperty>动作标记可以在 JSP 页面中设置 Bean 实例的属性，但必须保证 Bean 有对应的 setxxx 方法。

使用格式如下：

```
<jsp:setProperty name="Bean 对象的名称" property="属性名" value="<%= expression %>" />
```

上面题目中用到的：

```
<jsp:setProperty name="book" property="bookname" value="<%= bookname%>"/>
```

其中 name 值必须与<jsp:useBean>中 id 的值相同，表示类的对象；property 的值是进行设置的对象的属性值；表达式的数据类型必须和 Bean 的属性类型相一致。

<jsp:setProperty>命令有以下四种形式的用法。

(1) 第一种形式

```
<jsp:setProperty name = " Bean 对象的名称" property = " * "/>
```

该形式是设置 Bean 属性的快捷方式。在 Bean 中属性的名字、类型必须和 request 对象中的参数名称相匹配。由于表单中传过来的数据类型都是 String 类型的，JSP 内在机制会把这些参数转化成 Bean 属性对应的类型。

property = " * "表示所有名字和 Bean 属性名字匹配的请求参数都将被传递给相应的属性 set 方法。

(2) 第二种形式

```
<jsp:setProperty name = " Bean 对象的名称" property = " propertyName " />
```

使用 request 对象中的一个参数值来指定 Bean 中的一个属性值。在这个语法中，

property 指定 Bean 的属性名,而且 Bean 属性和 request 参数的名字应相同。也就是说,如果在 Bean 中有 setUserName(String userName)方法,那么,propertyName 的值就是"userName"。这种形式灵活性较强,可以有选择地对 Bean 中的属性赋值。

(3) 第三种形式

```
<jsp:setProperty name = " Bean 对象的名称" property = "Bean 对象的属性名" value =
"BeanValue"/>
```

value 用来指定 Bean 属性的值。字符串数据会在目标类中通过标准的 valueOf 方法自动转换成数字、boolean、Boolean、byte、Byte、char、Character。例如,boolean 和 Boolean 类型的属性值(比如 true)通过 Boolean. valueOf 转换,int 和 Integer 类型的属性值(比如 42)通过 Integer. valueOf 转换。

(4) 第四种形式

```
<jsp:setProperty name = " Bean 对象的名称" property = "propertyName" param = "request 对
象中的参数名"/>
```

param 指定用哪个请求参数作为 Bean 属性的值。Bean 属性和 request 参数的名字可以不同。如果当前请求没有参数,则什么事情也不做,系统不会把 null 传递给 Bean 属性的 set 方法。因此,你可以让 Bean 自己提供默认属性值,只有当请求参数明确指定了新值时才修改默认属性值。

3.6.4 <jsp:getProperty> 动作标记

使用<jsp:getProperty>动作标记可以在 JSP 页面中得到 Bean 实例的属性值,并将其转换为 String 类型的字符串,发送到客户端。使用<jsp:getProperty>动作标记,必须先声明 Bean 的实例,并保证 Bean 中有对应的 getXxx()方法。

语法格式如下:

```
<jsp:getProperty name="Bean 对象名称" property="属性名">
```

上面题目中用到的:

```
<jsp:getProperty name="book" property="bookname"/>
```

其中 name 值必须与<jsp:useBean>中的 id 的值相同,表示类的对象;property 的值是进行取值的对象的属性值。

实战演练 2——应用 JavaBean 实现管理员登录

【题目】

要求使用 useBean 动作完成本题。设计一个管理员登录的界面,如果管理员能够顺利登录就可以进入图书管理界面对图书进行入库操作;如果管理员登录失败就要重新回到登录界面,不能进入图书管理界面(不使用数据库)。

【分析】

使用 useBean 动作来完成此题,首先应该创建一个用于存放登录信息的 Bean,在这

个 Bean 中要包括登录的用户名和密码的 get 和 set 方法，以及用于鉴别登录信息是否正确的方法。本题需要两个界面来完成操作，一个界面用于登录；另一个界面用于显示。所以本题共需要三个文件，login. java(登录信息的 Bean)、login. html(登录界面)、login_show. jsp(登录结果显示界面)。

【运行效果】

首先创建 Bean，打开开发工具 MyEclipse，在工程 jsp-yufa 的 src 文件夹中已经建好的一个用于放置 Bean 的文件夹 bean 中新建类 login。在新建好的 login. java 文件中输入程序源代码，然后保存。当程序运行之后，login. java 文件会自动转换成 login. class 以供程序运作使用。

在工程 jsp-yufa 中的 WebRoot 文件夹下，新建工程文件 login. html 和 login_show. jsp。程序输入完毕并且保存之后，运行开发工具 MyEclipse 中的 Tomcat 服务器。最后在浏览器中输入地址 http://127. 0. 0. 1:8080/jsp-yufa/ login. html 就可以看到运行结果了(如图 3-27～图 3-29 所示)。



图 3-27 login. html 登录界面

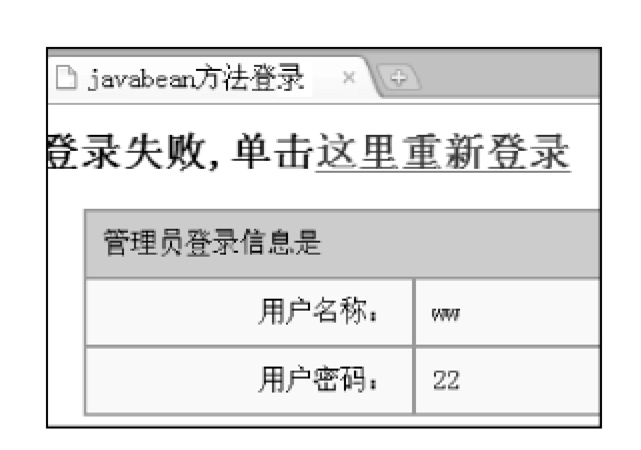


图 3-28 login_show. jsp 登录失败



图 3-29 login_show. jsp 登录成功

【源代码】

【login. java】

```
package bean;
public class login {
    private String loginName;
    private String password;
    public   void   login(){
        loginName=null;
        password=null;
    }
    public String getloginName()
    {return loginName;}
```

```
        public void setloginName(String loginName)
        {this.loginName = loginName;     }
        public String getpassword()
        {return password;}
        public void setpassword(String password)
        {this.password = password;     }
        public  boolean check()
        {
            if("sunqian".equals(loginName)&&"sq".equals(password))
                return true;
            else
            return false;
        }
}
```

【login.html】

```
<html>
  <head>
    <title>javabean方法登录</title>
  </head>
  <body>
    <form method="post" action="login_show.jsp" name="form1" >
      <table width="90%" border="1" align="center" cellpadding="10" cellspacing="0"
      bordercolor="#CC9900" bgcolor="#FFFFCC">
        <tr>
          <td  colspan="2" align="left" valign="middle" bgcolor="#CCCCCC" >管理
          员登录</td>
          </tr>
          <tr>
            <td width="40%" align="right" valign="middle">用户名称：</td>
            <td align="left" valign="middle">
              <input type="text" name="loginName" size="20"/></td>
          </tr>
          <tr>
            <td align="right" valign="middle" >用户密码：</td>
            <td align="left" valign="middle">
              <input type="password" name="password" size="20"/></td>
          </tr>
          <tr>
            <td colspan="2" align="center" valign="middle">
               <input type="submit" name="submit" value="提交" size="6" />
               <input type="reset"  name="reset" value="重置" size="6" /></td>
          </tr>
      </table>
    </form>
  </body>
</html>
```

【login_show.jsp】

```
<%@ page contentType="text/html;charset=GB2312" %>
<html>
```

```
  <head>
    <title>javabean 方法登录</title>
  </head>
  <body>
    <jsp:useBean id="manage" class="bean.login" scope="page" />
    <jsp:setProperty name="manage"  property="*" />
    <%if (manage.check())
    {%>
    <h2>欢迎管理员<jsp:getProperty name="manage" property="loginName"/>登录
    <a href="cart.html">到库存页面去工作!</a>
    </h2>
    <%
    }
    else
    {%>
    <h2>登录失败,单击<a href="login.html">这里重新登录</a></h2>
    <% }%>
    <p>
    <table width="90%" border="1" align="center" cellpadding="10" cellspacing="0" bordercolor="#CC9900" bgcolor="#FFFFCC">
    <tr>
      <td colspan="2" align="left" valign="middle" bgcolor="#CCCCCC">管理员登录信息是</td>
    </tr>
    <tr>
      <td align="right" valign="middle" width="40%">用户名称:</td>
      <td align="left" valign="middle"><jsp:getProperty name="manage" property="loginName"/></td>
    </tr>
    <tr>
      <td align="right" valign="middle" >用户密码:</td>
      <td align="left" valign="middle"><jsp:getProperty name="manage" property="password"/></td>
    </tr>
    </table>
    <p align="center"><a href="js.jsp">返回到访问计数器页面</a>
    </body>
</html>
```

【补充说明】

login.java 代码说明

第 1 行表示 login.java 在 bean 这个文件夹中；第 2 行到第 24 行是一个完整的 login 类的声明，login 类是一个公共类，类的名字是自动生成的并且与文件名相同；第 3 行到第 4 行是 login 类中的两个私有变量，loginName 表示管理员用户名，password 表示登录密码；第 5 行到第 8 行是 login 类的构造函数，是不带参数的，用于对 login 的对象进行初始化；第 9 行到第 10 行是获取管理员用户名的方法，可以返回管理员名字的字符串；第 11 行到第 12 行是设置管理员用户名的方法；第 13 行到第 14 行是获取登录密码的方法，可以返回密码的字符串；第 17 行到第 23 行是一个返回值为布尔值的方法，用于判断登

录界面输入的用户名和密码是否符合管理员的默认值，如果正确则返回值为 true，否则返回值为 false。

login.html 代码说明

本页面是登录界面的设计。第 6 行到第 25 行是一个表单，表单中的数据将要提交到 login_show.jsp 代码页；第 7 行到第 24 行是一个表格，表格总共有 4 行 2 列；第 8 行到第 10 行是表格的第一行，这行把两个单元格合并为 1 个，并且设置了该行的格式和颜色等属性；第 11 行到第 14 行是表格的第二行，这行的第二个单元格放置了一个文本框，用于接收输入的管理员用户名，变量名为 loginName；第 15 行到第 18 行是表格的第三行，这行的第二个单元格放置了一个密码框，用于接收登录密码，变量名为 password；第 19 行到第 23 行是表格的第四行，这行的两个单元格合并为 1 个，放置了两个按钮，一个用于提交表单的数据；另一个用于表格内容的重新输入。

login_show.jsp 代码说明

第 7 行表示使用 useBean 动作，在此动作中用到的类是 bean 包下的 login 类，建立类的对象是 manage，使用范围是 page；第 8 行表示设置对象 manage 的属性值为 login.html 表单传递过来的管理员用户名和密码，property = "*"表示所有和 manage 对象属性名字匹配的参数都将被传递给相应的 set 方法，本题使用这个快捷方法是因为表单传递的参数值都是 String 类型，正好与 login 类中属性的类型是符合的；第 9 行到第 19 行是一个 if 条件语句，通过调用 manage 对象的 check()方法来判断 login.html 登录界面输入的用户名和登录密码是否正确，如果都正确就输出带有管理员登录名字的欢迎语句和去库存页面的链接，如果不正确，就给出重新登录的返回链接；第 21 行到第 33 行是一个表格，表格分为 3 行 2 列，无论登录信息正确与否，都会在第二行第二个单元格和第三行第二个单元格显示登录界面输入过的用户名和密码。

实战演练 3——应用 JavaBean 实现网站访问计数器

【题目】

本题要求制作一个网站访问计数器，当同一个用户进行页面刷新或页面跳转的时候不能重复计数。

【分析】

本题需要两个 Bean，一个是负责计数，命名为 save；另一个负责检查是否为首次访问，命名为 fw。在 save 类中应该有获取访问量的 get 方法和实现访问量累计的方法。fw 类中应该有一个布尔变量用于判断用户是否为首次访问，还应该有一个方法根据用户是否为首次访问的结果来执行操作，如果为首次访问就调用 save 类中的方法进行访问量累计，否则直接调用 save 类中的 get 方法输出已有的访问量。

本题的界面只需要一个，界面需要显示访问量，以及提供前面两个例题的链接。

【运行效果】

首先创建 Bean，打开开发工具 MyEclipse，在工程 jsp-yufa 的 src 文件夹中已经建好的一个用于放置 Bean 的文件夹 bean 中新建类 save 和 fw。在新建好的 save.java 和

fw.java 文件中输入程序源代码然后保存。

在工程 jsp-yufa 中的 WebRoot 文件夹下，新建工程文件 js.jsp。程序输入完毕并且保存之后，运行开发工具 MyEclipse 中的 Tomcat 服务器。最后在浏览器中输入地址 http://127.0.0.1:8080/jsp-yufa/js.jsp 就可以看到运行结果了（如图 3-30 所示）。

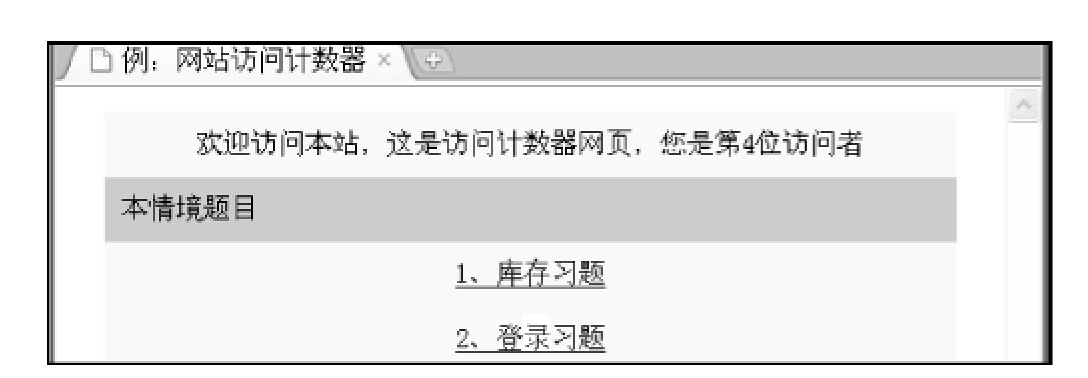


图 3-30　js.jsp 网站访问计数器

【源代码】

【save.java】

```
package bean;
public class save {
    private long counter=0;
    public long getCounter()
    {
        return counter;
    }
    synchronized public void Counter()
    {
        this.counter++;
    }
}
```

【fw.java】

```
package bean;
import bean.save;
public class fw {
    boolean isCome=false;
    public long getCount(save countB)
    {
        if(isCome==false)
        {
            countB.Counter();
            isCome=true;
        }
        return countB.getCounter();
    }
}
```

【js.jsp】

```
<%@page contentType="text/html;charset=GB2312" %>
<jsp:useBean id="b" class="bean.save" scope="application">
```

```
</jsp:useBean>
<jsp:useBean id="a" class="bean.fw" scope="session">
</jsp:useBean>
<html>
  <head>
    <title>例：网站访问计数器</title>
  </head>
  <body>
    <p>
      <%
        long num;
        num=a.getCount(b);
      %></p>
    <table width="500" border="0" align="center" cellpadding="10" cellspacing="0" bgcolor="#FFFFCC">
      <tr>
        <td align="center" >欢迎访问本站,这是访问计数器网页,您是第<%=num %>位访问者</td>
      </tr>
      <tr>
        <td bgcolor="#CCCCCC">本情境题目</td>
      </tr>
      <tr>
        <td align="center" ><a href="kucun.html">1、库存习题</a></td>
      </tr>
      <tr>
        <td align="center"><a href="login.html">2、登录习题</a></td>
      </tr>
    </table>
  </body>
</html>
```

【补充说明】

save.java 代码说明

第 1 行表示 save.java 在 bean 这个文件夹中；第 2 行到第 12 行是一个公共类 save 的声明；第 3 行是私有变量 counter 的声明并且被赋予了初始值，这个类里面没有构造函数，编译系统会自动插入一个无参数的默认构造函数，这个构造函数不执行任何代码；第 4 行到第 7 行是获取访问数的方法；第 8 行到第 11 行是访问量计数累计方法，这里用 synchronized 来解决多个线程同时访问同一个共享变量的问题，也就是说当同一个用户在本网站访问时是不会因为刷新了页面就做访问数量的累计的。

fw.java 代码说明

第 1 行表示 fw.java 在 bean 这个文件夹中；第 2 行引入 save 类；第 3 行到第 14 行是一个公共类 fw 的声明；第 4 行是私有变量 isCome 的声明，并且被赋予了初始值，这个类里面也没有构造函数；第 5 行到第 13 行是获取访问数的方法；第 7 行到第 11 行是通过布尔变量 isCome 的值判断用户是否为首次登录，如果是首次登录，那么就调用 save 类中的 Counter()方法进行计数累加；如果不是首次登录就直接返回 save 类中的 getCounter()方法的值。

js.jsp 代码说明

第 2 行表示使用 useBean 动作，在此动作中用到的类是 bean 包下的 save 类，建立类的对象是 b，使用范围是 application，这表示当不同的客户访问网站的时候，对网站访问产生的累计会延续下去，这样才能保证不同客户访问网站时的计数累加；第 3 行表示使用 useBean 动作的结束；第 4 行表示使用 useBean 动作，在此动作中用到的类是 bean 包下的 fw 类，建立类的对象是 a，使用范围是 session，这意味着客户在这个站点内刷新页面或者页面的跳转 b 的值都会保持不变；第 5 行表示使用 useBean 动作的结束；第 12 行到第 15 行是一个程序片，声明了一个长整形的变量 num，并给它赋值为 a.getCount(b)，当网站被首次访问时 a.getCount(b)就相当于对访问者做了访问量的累计；第 18 行用一个表达式输出了 num 的值；第 24 行和第 27 行是对前面两个习题的链接。

本 章 小 结

掌握 JSP 语法是学好 JSP 开发的基础。JSP 网页包括 HTML 标签和 Java 程序片，因此在 Java 中使用的各种数据类型在 JSP 中均可使用。Java 程序片用<%和%>括起来。在程序中加入注释可增强程序的可读性与可理解性，JSP 中的注释分为 HTML 注释和 JSP 注释两种。

本章给大家介绍的 JSP 指令标签有 page 指令、include 指令和 taglib 指令。Page 指令定义 JSP 页面的全局属性；include 指令用来静态插入一个文件；taglib 指令用于声明使用自定义的标签。

JSP 的动作较多，有 include、forward、param、useBean、setProperty、getProperty。Include 动作用来在 JSP 页面中动态包含一个文件，包括的文件可以是动态文件，也可以是静态文件；forward 动作用来重定向网页；param 动作配合 include 动作和 forward 动作使用，用来设置这两个动作的参数值；useBean 是指在 JSP 中创建并使用 JavaBean 技术；setProperty 和 getproperty 分别用来设置和获得 bean 的属性值。

中文字符处理是 JSP 程序员常常感到头痛的问题，如果把获得的字符串转换为一个 byte 数组，再转换为字符串，就可以正确地显示中文了。

本 章 习 题

1. 下面习题是关于表单信息提交并且实现汉字识别的，按照下面的源代码在浏览器中做出如图 3-31 和图 3-32 所示的结果。

【源代码】

【book.jsp】

```
<%@ page contentType="text/html;charset=gb2312"
language="java" %>
<html>
```

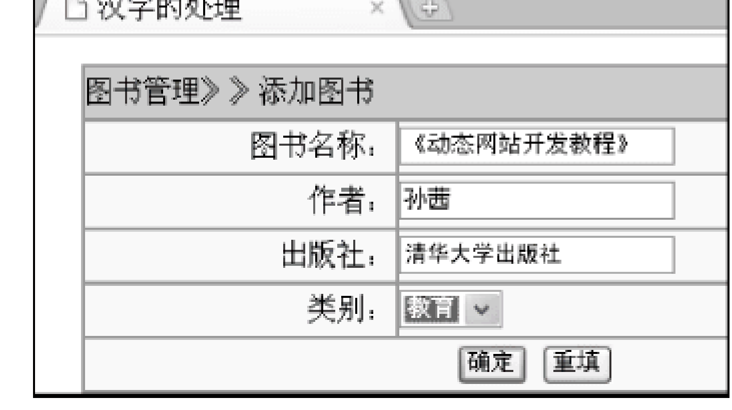


图 3-31 book.jsp



图 3-32　book_show.jsp

```
<head>
  <title>汉字的处理</title>
</head>
<body>
  <form method="post" action="book_show.jsp" name="form1">
      <table width="500" border="1" align="center" cellpadding="0" cellspacing="0"
      bordercolor="#CC9900" bgcolor="#FFFFCC">
        <tr>
          <td height="30" colspan="2" align="left" valign="middle" bgcolor="#CCCCCC">
          图书管理》》添加图书</td>
      </tr>
      <tr>
        <td height="30" align="right" valign="middle">图书名称：</td>
        <td height="30" align="left" valign="middle">
          <input type="text" name="bookname" id="textfield2" /></td>
      </tr>
      <tr>
        <td height="30" align="right" valign="middle" >作者：</td>
        <td height="30" align="left" valign="middle">
          <input type="text" name="author" id="textfield3" /></td>
      </tr>
      <tr>
        <td height="30" align="right" valign="middle">出版社：</td>
        <td height="30" align="left" valign="middle">
          <input type="text" name="publisher" id="textfield4" /></td>
      </tr>
      <tr>
        <td height="30" align="right" valign="middle">类别：</td>
        <td height="30" align="left" valign="middle">
          <select name="category" id="select">
            <option selected="selected">科技</option>
            <option>艺术</option>
            <option>生活</option>
            <option>社会</option>
            <option>娱乐</option>
            <option>教育</option>
          </select></td>
      </tr>
      <tr>
        <td height="30" colspan="2" align="center" valign="middle">
          <input type="submit" name="button" id="button" value="确定" />
          <input type="reset" name="button2" id="button2" value="重填" />
        </td>
```

```
        </tr>
      </table>
    </form>
  </body>
</html>
```

【book_show.jsp】

```
<%@ page contentType="text/html;charset=gb2312" language="java" %>
<%@ include file="hzsb.jsp" %>
<html>
  <head>
    <title>汉字的处理</title>
  </head>
  <body>
    <%
        String bookname=request.getParameter("bookname");
        String author=request.getParameter("author");
        String publisher=request.getParameter("publisher");
        String category=request.getParameter("category");
        bookname=hzsb(bookname);
        author=hzsb(author);
        publisher=hzsb(publisher);
        category=hzsb(category);
     %>
     <table width="700" border="1" align="center" cellpadding="0" cellspacing="0"
     bordercolor="#CC9900" bgcolor="#FFFFCC">
          <tr>
            <td height="30" align="center" valign="middle" bgcolor="#CCCCCC">图书
            名称  </td>
            <td height="30" align="center" valign="middle" bgcolor="#CCCCCC">作者 &
            nbsp;</td>
            <td height="30" align="center" valign="middle" bgcolor="#CCCCCC">出版
            社  </td>
            <td height="30" align="center" valign="middle" bgcolor="#CCCCCC">类别 &
            nbsp;</td>
          </tr>

          <tr>
            <td height="30" align="center" valign="middle">
                 <%=bookname %> </td>
            <td height="30" align="center" valign="middle">
                 <%=author %> </td>
            <td height="30" align="center" valign="middle">
                 <%=publisher %> </td>
            <td height="30" align="center" valign="middle">
                 <%=category %> </td>
          </tr>
        </table>
  </body>
</html>
```

【hzsb. jsp】

```
<%@ page contentType="text/html;charset=gb2312" language="java" %>
<%!
    public String hzsb(String s)
    {
        String str=s;
        try
        {
            byte b[]=str.getBytes("ISO-8859-1");
            str=new String(b);
            return str;
        }
        catch(Exception e)
        {return str;}
    }
%>
```

2. 下面习题是学生总评成绩计算器，可以输入学生的班级、姓名、8 次平时成绩、考勤成绩、期末成绩，单击“计算”按钮可以得出结果，如图 3-33 和图 3-34 所示。



图 3-33 score. jsp

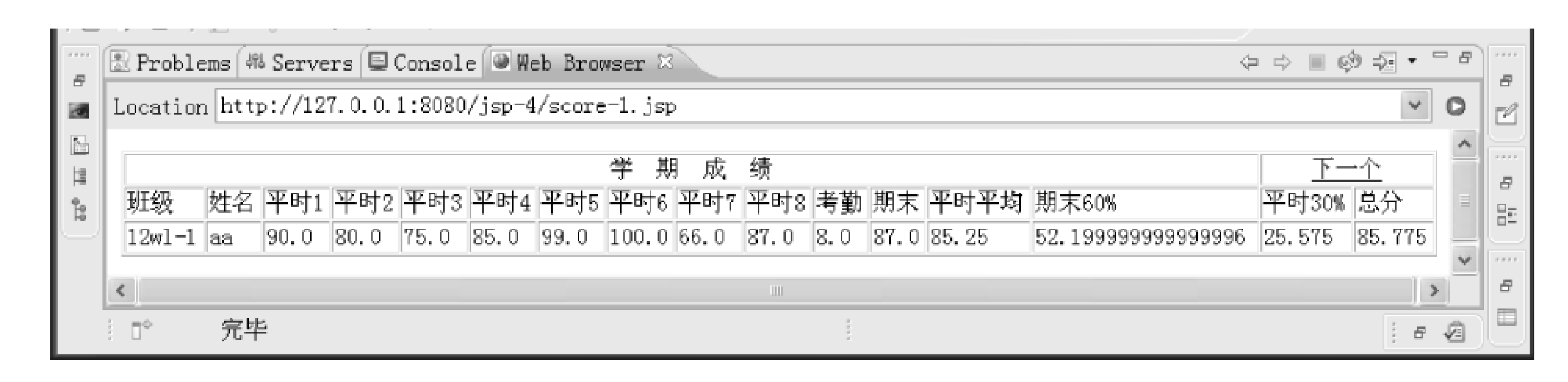


图 3-34 score-1. jsp

【源代码】

【score. java】

```
package bean;
public class score {
    private String bj;
    private String xm;
    private Double p1;
    private Double p2;
```

```
private Double p3;
private Double p4;
private Double p5;
private Double p6;
private Double p7;
private Double p8;
private Double kq;
private Double qm;
private double ave=0.0;
private double ave1=0.0;
private double qm1=0.0;
public void setBj(String bj)
{   this.bj=bj;   }
public void setXm(String xm)
{   this.xm=xm;   }
public void setP1(Double p1)
{   this.p1=p1;   }
public void setP2(Double p2)
{   this.p2=p2;   }
public void setP3(Double p3)
{   this.p3=p3;   }
public void setP4(Double p4)
{   this.p4=p4;   }
public void setP5(Double p5)
{   this.p5=p5;   }
public void setP6(Double p6)
{   this.p6=p6;   }
public void setP7(Double p7)
{   this.p7=p7;   }
public void setP8(Double p8)
{   this.p8=p8;   }
public void setKq(Double kq)
{   this.kq=kq;   }
public void setQm(Double qm)
{   this.qm=qm;   }
public String getBj()
{   return(bj); }
public String getXm()
{   return(xm); }
public Double getP1()
{   return(p1); }
public Double getP2()
{   return(p2); }
public Double getP3()
{   return(p3); }
public Double getP4()
{   return(p4); }
public Double getP5()
{   return(p5); }
```

```
        public Double getP6()
        {   return(p6); }
        public Double getP7()
        {   return(p7); }
        public Double getP8()
        {   return(p8); }
        public Double getKq()
        {   return(kq); }
        public Double getQm()
        {   return(qm); }
        public double average()
        {
            ave=(p1+p2+p3+p4+p5+p6+p7+p8)/8;
            return(ave);
        }
        public double qimo()
        {
            qm1=qm * 0.6;
            return(qm1);
        }
        public double pingshi()
        {
            ave1=ave * 0.3;
            return(ave1);
        }
        public double sum()
        {
            double sum=kq+ave1+qm1;
            return(sum);
        }
    }
```

【score.jsp】

```
<%@ page contentType="text/html;charset=GB2312" %>
<html>
  <head>
        <title>My JSP 'score.jsp' starting page</title>
    </head>
  <body>
    <form name="form1" method="post" action="score-1.jsp">
      <table width="900" border="0" cellspacing="0" cellpadding="0">
        <tr>
          <td colspan="13" align="center">学   期   成  
           绩</td>
        </tr>
        <tr><td colspan="13" align="center"> </td></tr>
        <tr>
          <td>班级</td>
          <td>姓名</td>
```

```
            <td>平时 1</td>
            <td>平时 2</td>
            <td>平时 3</td>
            <td>平时 4</td>
            <td>平时 5</td>
            <td>平时 6</td>
            <td>平时 7</td>
            <td>平时 8</td>
            <td>考勤</td>
            <td>期末</td>
            <td> </td>
          </tr>
          <tr>
            <td><input name="bj" type="text"   size="5" ></td>
            <td><input name="xm" type="text"   size="5" ></td>
            <td><input name="p1" type="text"   size="5" ></td>
            <td><input name="p2" type="text"   size="5"></td>
            <td><input name="p3" type="text"   size="5"></td>
            <td><input name="p4" type="text"   size="5"></td>
            <td><input name="p5" type="text"   size="5"></td>
            <td><input name="p6" type="text"   size="5"></td>
            <td><input name="p7" type="text"   size="5"></td>
            <td><input name="p8" type="text"   size="5"></td>
            <td><input name="kq" type="text"   size="5"></td>
            <td><input name="qm" type="text"   size="5"></td>
            <td><input type="submit" name="button" id="button" value="计算"></td>
          </tr>
        </table>
      </form>
    </body>
  </html>
```

【score-1.jsp】

```
  <%@ page contentType="text/html;charset=GB2312" %>
  <html>
    <head>
          <title>My JSP 'score.jsp' starting page</title>
      </head>
    <body>
      <jsp:useBean id="a" class="bean.score" scope="page" />
      <jsp:setProperty name="a"   property=" * "   />
      <table width="900px" border="1" cellspacing="1" cellpadding="1">
        <tr>
          <td colspan="14" align="center">学   期   成  
           绩</td>
          <td colspan="2" align="center"><a href="score.jsp">下一个</a></td>
        </tr>
        <tr>
          <td>班级</td>
```

```
        <td>姓名</td>
        <td>平时 1</td>
        <td>平时 2</td>
        <td>平时 3</td>
        <td>平时 4</td>
        <td>平时 5</td>
        <td>平时 6</td>
        <td>平时 7</td>
        <td>平时 8</td>
        <td>考勤</td>
        <td>期末</td>
        <td>平时平均</td>
        <td>期末 60%</td>
        <td>平时 30%</td>
        <td>总分</td>
      </tr>
      <tr>
        <td><jsp:getProperty name="a" property="bj"/></td>
        <td><jsp:getProperty name="a" property="xm"/></td>
        <td><jsp:getProperty name="a" property="p1"/></td>
        <td><jsp:getProperty name="a" property="p2"/></td>
        <td><jsp:getProperty name="a" property="p3"/></td>
        <td><jsp:getProperty name="a" property="p4"/></td>
        <td><jsp:getProperty name="a" property="p5"/></td>
        <td><jsp:getProperty name="a" property="p6"/></td>
        <td><jsp:getProperty name="a" property="p7"/></td>
        <td><jsp:getProperty name="a" property="p8"/></td>
        <td><jsp:getProperty name="a" property="kq"/></td>
        <td><jsp:getProperty name="a" property="qm"/></td>
        <td><%=a.average() %></td>
        <td><%=a.qimo() %></td>
        <td><%=a.pingshi() %></td>
        <td><%=a.sum() %></td>
      </tr>
    </table>
  </body>
</html>
```

情境四

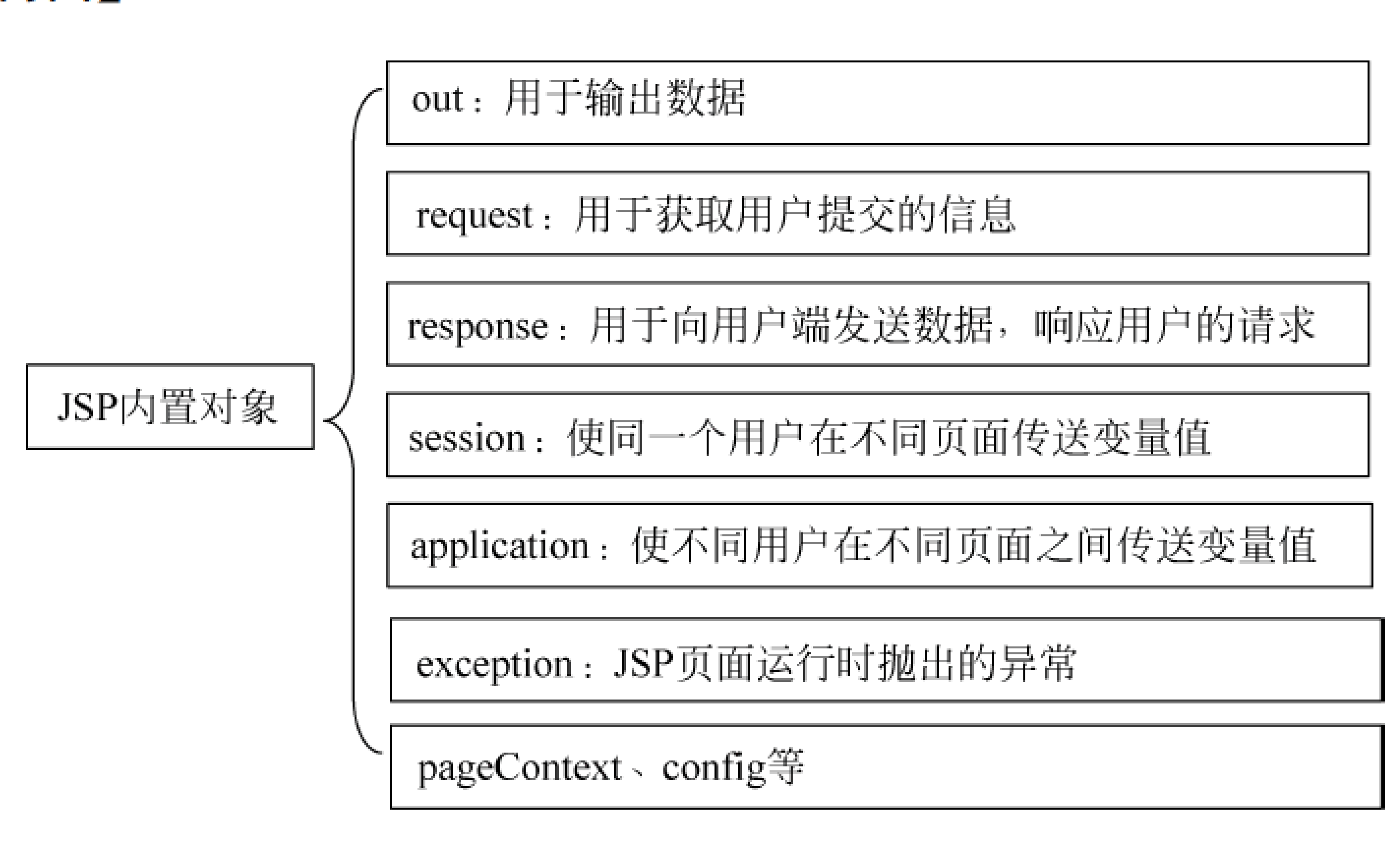
JSP 内置对象

Project 4

【学习目标】

1. 了解表单验证的知识。
2. 掌握 request 对象获得表单数据的方法。
3. out 输出各种格式数据的方法。
4. response 对象响应等方法。
5. session 对象的生命周期和保存、获取数据的方法。
6. 掌握 application 对象数据的保存与获取方法。

【知识结构图】



JSP 内置对象概述

JSP 的内置对象是指 JSP 页面系统中已经默认设置的 Java 对象，这些对象无须显式声明，直接就可以在 Java 程序片和表达式中使用。

JSP 的内置对象有 request、response、session、application、out、config、exception、page 和 pageContext 对象，这些对象分别完成不同的任务。

request 对象封装了客户提交的数据信息及客户端的一些信息，如客户端使用表单提交数据；response 对象用来对客户端请求做出响应，向客户端发送数据；session 对象用来保存需要在与每个用户会话期间保持的数据信息；application 对象用来在多个程序或者是多个用户之间共享数据；out 用来向客户端输出数据。

4.1　使用 request 对象

实战演练 1——request 计算器

【题目】

制作一个简单的实数计算器，可以完成加减乘除运算。

【分析】

本题需要做一个界面设计，我们假设对两个操作数进行计算，那么在界面上需要有两个文本框，用来输入操作数，四个单选按钮分别设计为不同的操作(加、减、乘、除)，一个提交按钮，还有一个接收操作结果的空间。本题操作结果的获得需要使用 request 对象的 getParameter 方法来实现页面间数据的接收。总的来说本题需要两个页面来完成操作，一个是操作界面 computer.jsp；另一个是后台计算的代码 cc.jsp。

【运行效果】

打开开发工具 MyEclipse，新建一个工程 jsp-nz，在这个工程的 WebRoot 文件夹下，新建工程文件 computer.jsp、cc.jsp。程序输入完毕并且保存之后，运行开发工具 MyEclipse 中的 Tomcat 服务器。最后在浏览器中输入地址 http://127.0.0.1:8080/jsp-nz/computer.jsp 就可以看到运行结果了(如图 4-1～图 4-3 所示)。

计算器

计算器				
第一个数	操作	第二个数	结果	
	○ + ○ − ○ × ○ ÷		=	null

图 4-1　computer.jsp 首次登录界面



图 4-2　computer.jsp 输入操作数并选择操作

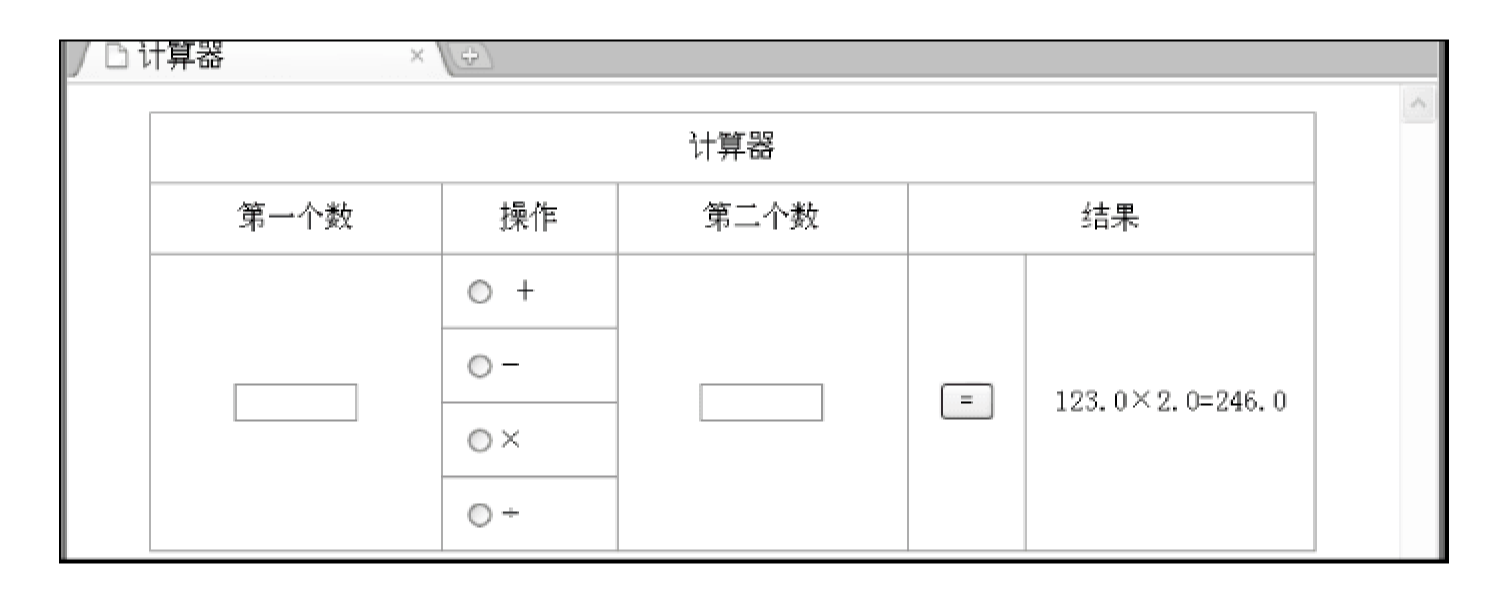


图 4-3　computer.jsp 得到结果

【源代码】

【computer.jsp】

```
<%@ page contentType="text/html; charset=gb2312" language="java" %>
<html>
  <head>
    <title>计算器</title>
  </head>
  <body>
    <form name="form1" method="post" action="cc.jsp">
      <table width="90%" border="0" align="center" cellpadding="10" cellspacing="1"
      bgcolor="#CC9900">
        <tr>
          <td colspan="5" align="center" bgcolor="#FFFFFF">计算器</td>
        </tr>
        <tr>
          <td width="25%" align="center" bgcolor="#FFFFFF">第一个数</td>
          <td width="15%" align="center" bgcolor="#FFFFFF">操作</td>
          <td width="25%" align="center" bgcolor="#FFFFFF">第二个数</td>
          <td width="40%" colspan="2" align="center" bgcolor="#FFFFFF">结果
          </td>
        </tr>
        <tr>
          <td rowspan="4" align="center" valign="middle" bgcolor="#FFFFFF">
            <input name="a" type="text" size="8" /></td>
          <td bgcolor="#FFFFFF">
            <input type="radio" name="cz" value="add"/>+</td>
          <td rowspan="4" align="center" valign="middle" bgcolor="#FFFFFF">
            <input name="b" type="text" size="8" /></td>
          <td rowspan="4" align="center" valign="middle" bgcolor="#FFFFFF">
            <input type="submit" name="button1" value=" = "/></td>
          <td width="30%" rowspan="4" align="center" valign="middle" bgcolor=
          "#FFFFFF">
            <%String result=request.getParameter("result"); %>
            <%=result %>
          </td>
        </tr>
        <tr>
```

```
33               <td bgcolor="#FFFFFF">
34                 <input type="radio" name="cz" value="sub"/>-</td>
35             </tr>
36             <tr>
37               <td bgcolor="#FFFFFF">
38                 <input type="radio" name="cz" value="mul"/>×</td>
39             </tr>
40             <tr>
41               <td bgcolor="#FFFFFF">
42                 <input type="radio" name="cz" value="div"/>÷</td>
43             </tr>
44           </table>
45         </form>
46     </body>
47 </html>
```

【cc.jsp】

```
<%@ page contentType="text/html; charset=gb2312" language="java" %>
<html>
  <body>
    <%   double a=Double.parseDouble(request.getParameter("a"));
       double b=Double.parseDouble(request.getParameter("b"));
       String cz=request.getParameter("cz");
       String result=null;
       if (cz.equals("add"))
         {result=a+"+"+b+"="+(a+b);
       if (cz.equals("sub"))
         {result=a+"-"+b+"="+(a-b);
       if (cz.equals("mul"))
         {result=a+"×"+b+"="+(a*b);
       if (cz.equals("div"))
         {result=a+"÷"+b+"="+(a/b);
  %>
    <jsp:forward page="computer.jsp">
    <jsp:param name="result" value="<%=result%>"/>
    </jsp:forward>
  </body>
</html>
```

【补充说明】

computer.jsp 代码说明

本页面是计算器的操作界面，第 7 行到第 45 行是一个表单，表单数据将提交到 cc.jsp 代码页执行；第 8 行到第 44 行是一个表格，总共有 6 行 5 列；第 9 到第 11 行是表格第一行，这行合并了 5 个单元格；第 12 到第 17 行是表格第二行，这行总共有四个单元格；第 18 到第 31 行是表格第三行，这行总共有五个单元格，分别放置了两个文本框用于输入操作数，变量名分别为 a 和 b，单元格大小为 8，放置了一个单选按钮，变量名为 cz，标识为 add，表示这是一个＋操作，一个提交按钮，按钮上的文字修改为＝，第五个单元格用于放置计算结果，使用 request 来接收操作结果并使用表达式输出显示到界面上；第

32 到第 35 行是表格的第四行，放置了一个单选按钮，变量名为 cz，标识为 sub，表示这是一个—操作；第 36 到第 39 行是表格的第五行，放置了一个单选按钮，变量名为 cz，标识为 mul，表示这是一个×操作；第 40 到第 43 行是表格的第六行，放置了一个单选按钮，变量名为 cz，标识为 div，表示这是一个÷操作。

cc.jsp 代码说明

本页面是对操作数进行计算的代码页，第 4 行和第 5 行是接收 computer.jsp 页面传递过来的两个操作数，并且强制转换为 Double 类型；第 6 行是对运算操作 cz 的接收；第 8 行到第 15 行是使用 if 条件语句对 cz 进行判断，然后根据匹配情况对两个操作数进行“加”、“减”、“乘”或“除”的运算；第 17 行是对页面的重定向，当操作数完成计算之后就把操作定位到 computer.jsp 页面显示结果；第 18 行是将本页面得到的结果传递到 computer.jsp 页面；第 19 行是 forward 动作的结束标志。

request 对象

request 对象封装了客户端提交的数据信息，包括用户提交的信息以及客户端的一些信息。客户端可以通过 HTML 表单或者在网页地址后面使用参数的方法提交数据，再用 request 对象的相关方法来获取提交的各种数据。

在 Web 程序设计中，用户一般是通过 HTML 表单提交信息的。表单的一般格式是：

```
<form method=get/post action="目的页面" name="表单名字" >
    提交手段部分
</form>
```

request 对象可以使用 getParameter(String s)方法获得该表单提交的信息。调用方法如下：

```
request.getParameter(String str)
```

其中，参数 str 是指定的参数名称，这个方法是 JSP 程序设计中最为常用的方法之一。

request 接收的表单提交信息大多数时候都是 String 类型的，常常需要做类型的转换，比如上面例题中用到的：

```
Double.parseDouble(request.getParameter("a"))
```

这句代码中 request 接收的 String 类型的值被转换为了 Double 类型。我们常用到的还有以下几种。

整形：Integer.parseInt()；

浮点型：Float.parseFloat()；

long 型：Lang.parseLang()。

实战演练 2——request 选择题

【题目】

制作一篇测试题，由若干单选题和多选题组成，题目完成后提交答案，显示得分。

【分析】

本题需要两个界面，一个用于制作测试题(ex1.jsp)；另一个用于处理题目的正确率，得出分数(ex1-answer.jsp)。在 ex1.jsp 中需要添加表单、表格、若干单选和多选的按钮、提交按钮；在 ex1-answer.jsp 中需要使用 request 接收测试题页面 ex1.jsp 表单提交的数据，然后判断对错，得出分数。

【运行效果】

打开开发工具 MyEclipse，在上一个新建的工程 jsp-nz 中的 WebRoot 文件夹下，新建工程文件 ex1.jsp、ex1-answer.jsp。程序输入完毕并且保存之后，运行开发工具 MyEclipse 中的 Tomcat 服务器。最后在浏览器中输入地址 http://127.0.0.1:8080/jsp-nz/ex1.jsp 就可以看到运行结果了(如图 4-4 和图 4-5 所示)。

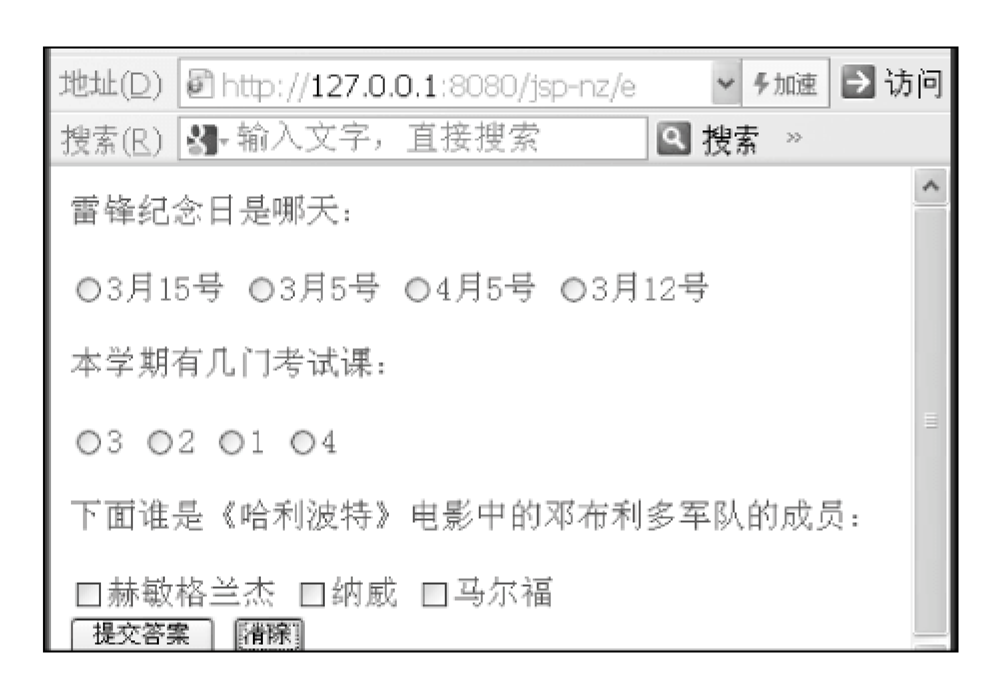


图 4-4 ex1.jsp 测试题

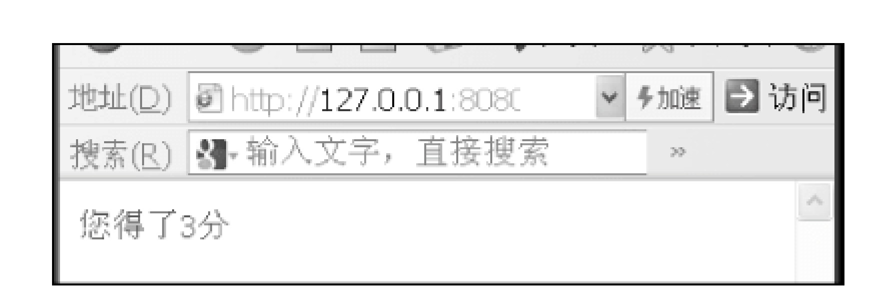


图 4-5 ex1-answer.jsp 得分情况

【源代码】

【ex1.jsp】

```
<%@ page contentType="text/html;charset=GB2312"%>
<html>
  <head>
    <title>选择题 request 练习</title>
  </head>
  <body>
    <form action="ex1-answer.jsp" method="post">
      <font color="#336699"size="4">
        <p>雷锋纪念日是哪天：  </p>
        <input type="radio" name="d1" value="a">3 月 15 号
        <input type="radio" name="d1" value="b">3 月 5 号
        <input type="radio" name="d1" value="c">4 月 5 号
        <input type="radio" name="d1" value="d">3 月 12 号
        <br>
      <p>本学期有几门考试课：  </p>
        <input type="radio" name="d2" value="a">3
        <input type="radio" name="d2" value="b">2
        <input type="radio" name="d2" value="c">1
        <input type="radio" name="d2" value="d">4
        <br>
```

```
21        <p>下面谁是《哈利波特》电影中的邓布利多军队的成员：  </p>
22          <input type="checkbox" name="c1" >赫敏格兰杰
23          <input type="checkbox" name="c2">纳威
24          <input type="checkbox" name="c3">马尔福
25          <br>
26          <input type="submit" value="提交答案" name="submit">
27          <input type="reset" value="清除">
28        </font>
29      </form>
30    </body>
31  </html>
```

【ex1-answer.jsp】

```
<%@ page contentType="text/html;charset=GB2312"%>
<html>
  <body>
    <font color="#336699"size="4">
      <%int n=0;
        String d1=request.getParameter("d1");
        String d2=request.getParameter("d2");
        String c1=request.getParameter("c1");
        String c2=request.getParameter("c2");
        String c3=request.getParameter("c3");
        if(d1==null)
        {d1="";}
        if(d2==null)
        {d2="";}
        if(d1.equals("b"))
        {n++;}
        if(d2.equals("a"))
        {n++;}
        if(!(c1==null)&&!(c2==null))
        {n++;}
      %>
    <p>您得了<%=n %>分</p>
  </body>
</html>
```

【补充说明】

ex1.jsp 代码说明

本页面是选择题页面，第 7 行到第 29 行是 1 个表单，表单的数据将提交到 ex1-answer.jsp 页面；第 8 行设置了页面中的字体格式和字体大小；第 9 行是段落标志；第 10 到第 13 行是在页面中添加了 4 个单选按钮，名称为 d1，标识分别为 a、b、c、d，代表单选题四个不同的选项；第 16 到第 19 行是在页面中添加了 4 个单选按钮，名称为 d2，标识分别为 a、b、c、d，代表单选题四个不同的选项；第 22 到第 24 行是在页面中添加了 3 个多选按钮，名称分别为 c1、c2、c3，代表单选题三个不同的选项；第 26 行是提交按钮；第 27 行是重置按钮。

ex1-answer. jsp 代码说明

本页面是得分页面，第 5 行，声明一个变量 n，用于累计得分；第 6 行到第 10 行是接收 ex1. jsp 页面传递过来的信息；第 11 行到第 14 行表示如果前面两个单选题没做，那么就给答案赋空值，这样可以有效预防选择题没做而出现的程序错误；第 15 行到第 19 行是判断 ex1. jsp 页面传递过来的信息是否与标准答案符合，如果符合就可以加分；第 22 行是以表达式输出得分。

4.2　response 对 象

实战演练 1——使用 response 实现重定向

【题目】

实现页面的重定向，分别使用 response 的无参数法、有参数法和 forward 方法。

【分析】

题目需要三个文件来完成，一个界面设计文件 response. jsp；一个后台设置不同方式完成页面重定向的文件 rr. jsp；一个显示重定向结果的页面 response_1. jsp。在 response. jsp 文件中需要一个表单，放入三个单选按钮，一个提交按钮，对不同的重定向方法都设置好标识；在 rr. jsp 文件中不需要界面的显示设计，只负责根据 response. jsp 文件传递过来的页面重定向方法的标识来设置具体的指令，并且为能够在页面重定向的时候携带参数的指令准备好参数值；response_1. jsp 文件就是最终被重定向显示的文件了，根据不同的重定向指令，显示不同的内容。

【运行效果】

打开开发工具 MyEclipse，在工程 jsp-nz 中的 WebRoot 文件夹下，新建工程文件 response. jsp、rr. jsp、response_1. jsp。程序输入完毕并且保存之后，运行开发工具 MyEclipse 中的 Tomcat 服务器。最后在浏览器中输入地址 http://127. 0. 0. 1:8080/jsp-nz/ response. jsp 就可以看到运行结果了（如图 4-6～图 4-9 所示）。

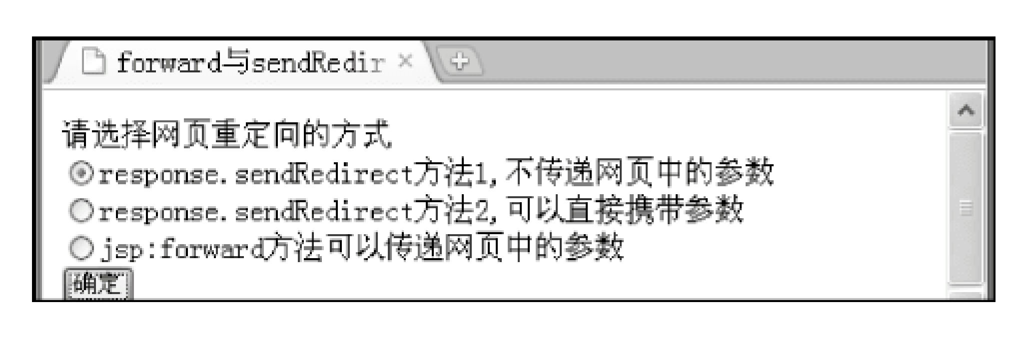


图 4-6　首页 response. jsp

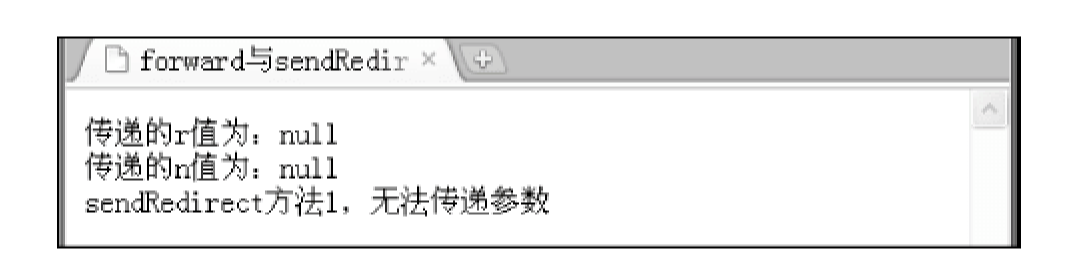


图 4-7　选择了 response 的无参数传递重定向

forward与sendRedir
传递的r值为：null
传递的n值为：null
sendRedirect方法1，无法传递参数，但是sendRedirect方法2，语句可以直接携带参数
sendRedirect方法2，携带的m值为：20

图 4-8　选择了 response 的有参数传递重定向

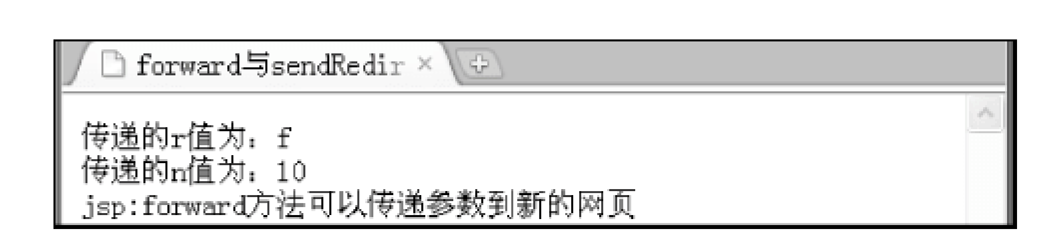


图 4-9　选择了 forward 的有参数传递重定向

【源代码】

【response.jsp】

```
<%@ page contentType="text/html;charset=gb2312" language="java" %>
<html>
  <head>
    <title>forward 与 sendRedirect 的比较</title>
  </head>
  <body>
    <form action="rr.jsp" method="post" >
    请选择网页重定向的方式<BR>
      <input type="radio" name="r" value="s1" CHECKED>response.sendRedirect 方法
      1,不传递网页中的参数<BR>
      <input type ="radio" name ="r" value="s2" >response.sendRedirect 方法 2,可以直
      接携带参数<BR>
      <input type ="radio" name ="r" value="f" >jsp:forward 方法可以传递网页中的参
      数<BR>
      <input type ="submit" name ="submit" value="确定">
    </form>
  </body>
</html>
```

【rr.jsp】

```
<%@ page contentType="text/html;charset=gb2312" language="java" %>
<html>
  <head>
    <title>My JSP 'rr.jsp' starting page</title>
  </head>
  <body>
    <%
      String r=request.getParameter("r");
      request.setAttribute("n","10");
      if(r.equals("s1"))
       {
       response.sendRedirect("response_1.jsp");
       }
       if(r.equals("s2"))
       {
       response.sendRedirect("response_1.jsp?m=20");
       }
      if(r.equals("f"))
      {
    %>
       <jsp:forward page="response_1.jsp">
       <jsp:param name="r" value="<%=r %>"/>
       </jsp:forward>
    <%
      }
    %>
```

```
27    </body>
28 </html>
```

【response_1.jsp】

```
<%@ page contentType="text/html;charset=gb2312" language="java" %>
<html>
  <head>
    <title>forward 与 sendRedirect 的比较</title>
  </head>
  <body>
    <%
      String n=(String)request.getAttribute("n");
      String r=(String)request.getParameter("r");
      String m=(String)request.getParameter("m");
      out.println("传递的 r 值为: "+r+"<br/>");
      out.println("传递的 n 值为: "+n+"<br/>");
      if(r!=null)
      {
      out.print("jsp:forward 方法可以传递参数到新的网页");
      }
      Else
      {
      out.print("sendRedirect 方法 1,无法传递参数");
      }
      if(m!=null)
      {
      out.print(",但是 sendRedirect 方法 2,语句可以直接携带参数"+"<br/>");
      out.println("sendRedirect 方法 2,携带的 m 值为: "+m+"<br/>");
      }
    %>
  </body>
</html>
```

【补充说明】

response.jsp 代码说明

本页面为界面设计,第 7 行到第 13 行是一个表单,表单中的数据将提交到 rr.jsp 页面;第 9 行放置了一个单选按钮,变量名为 r,标识设置为 s1,默认为选中状态,代表了 response 对象不带参数的页面重定向方法;第 10 行放置了一个单选按钮,变量名为 r,标识设置为 s2,代表了 response 携带参数的页面重定向方法;第 11 行放置了一个单选按钮,变量名为 r,标识设置为 f,代表了 forward 指令页面重定向方法;第 12 行为一个提交按钮。

rr.jsp 代码说明

本页面是根据 r 值的不同来确定重定向到 response_1.jsp 页面的方法。第 8 行是接收 response.jsp 页面传递过来的 r 值;第 9 行是设置 n 的值为 10;第 10 行到第 13 行是一个 if 条件语句,如果 r 值为 s1,就使用 response 无参数法定向到 response_1.jsp 页面;第 14 行到第 17 行是一个 if 条件语句,如果 r 值为 s2,,就使用 response 携带参数法定向到 response_1.jsp 页面,传递参数 m=20 的参数值;第 18 行到第 25 行是一个 if 条件语

句，如果 r 值为 f，就使用 forward 方法定向到 response_1.jsp 页面。

response_1.jsp 代码说明

本页面根据不同的页面定向方法，显示出不一样的结果。第 8 行到第 10 行是接收 rr.jsp 页面传递过来的参数；第 11 行到第 12 行是输出 r 的值和输出 n 的值，在 rr.jsp 页面 n 的值被 request 的 set 方法设置了指定值，如果使用 forward 指令做页面重定向就可以使用 request 的 get 方法再获得 n 的指定值；第 13 行到第 19 行是一个 if 条件语句，如果 r 的值不为空，那么就表示参数值是使用 forward 方法定向过来的，如果 r 的值为空，那么就表示参数值是使用 response 方法定向过来的；第 20 行到第 23 行是一个 if 条件语句，如果 m 的值不为空，那么就表示参数值是使用 response 的带参数发传递过来的。

4.2.1 out 对象

Out 对象实际上是使用 PrintWriter 类向客户端浏览器输出数据。

Out 对象的常用方法列举如下。

print()与 println()方法应用广泛，可以输出各种类型的数据。

如上题中使用到的：

out.print("，但是 sendRedirect 方法 2，语句可以直接携带参数"+"
")；

out.println("sendRedirect 方法 2，携带的 m 值为："+m+"
")。

4.2.2 response 对象

response 对象用于动态响应客户端的请求，将 JSP 处理数据后的结果返回到客户端浏览器。response 是 HttpServletResponse 类的一个对象，它与 request 对象对应。

1. 重定向 sendRedirect

sendRedirect 方法可以将客户端浏览器重定向到新的 Web 页面。使用 sendRedirect 可以根据用户的不同情况而定位到不同的页面上去，重定向语句之后的代码将不再执行。例如，下面语句就是停止当前网页执行，转向执行程序文件 response_1.jsp：

```
response.sendRedirect("response_1.jsp");
```

在前面介绍过<jsp：forward >也可以重定向到另一个页面，但两种操作有如下区别。

(1) 当使用<jsp：forward >操作重定向到另一个页面，可以通过<jsp:param>传递参数到新的网页，新的网页使用 request 对象的 getParameter()方法可以读取参数的值。而使用 response 对象的 sendRedirect()方法，不需要直接编码 URL，就可以实现传递参数。例如下面语句在重定向时传递了一个名为 m 值为 20 的参数，同样使用 request 对象访问参数 m。

```
response.sendRedirect("response_1.jsp?m=20");
…
String m=(String)request.getParameter("m");
```

(2) 当使用<jsp：forward >操作，两个页面之间可以通过 request 对象分享数据，而

response 对象的 sendRedirect()方法不支持这种数据共享。也就是说,使用 forward 时,request 设置的属性(setAttribute)依然能保留到下一个页面。就如同上面的例题中先在 rr.jsp 页面使用 set 方法设置了 n 的值为 10,然后当使用 forward 方法跳转页面时,在 response_1.jsp 页面使用 get 方法获取了 n 的值,如图 4-9 所示。

```
request.setAttribute("n","10");
…
String n=(String)request.getAttribute("n");
```

2. 修改 ContentType 属性

contentType 属性用来设置 JSP 页面的 MIME 类型和字符编码集,取值格式为"MIME 类型"或"MIME 类型;charset=字符编码集"。JSP 引擎根据 contentType 属性,对用户的请求做出响应。response 对象可以调用 setContentType(String s)方法动态修改 contentType 属性值,JSP 引擎会按照修改后的 MIME 类型来响应客户浏览器。MIME 的中文名为多功能网络邮件扩展标准,也就是说 MIME 是服务器通知客户端接收文件类型的一种手段。对不同 MIME 类型的文件,客户端会指定不同程序或插件打开,如 flash 动画指定其播放插件来播放。如下面例题中用到的语句会指定 Word 软件来打开并保存当前页面。

```
response.setContentType("application/msword;charset=GB2312");
```

3. 定时刷新页面

HTTP 响应消息主要由响应行、响应头和信息体组成。response 对象的 addHeader 方法或者 setHeader 方法可以动态添加新的响应头和头值,如果这个头事先已经存在,则原来的头被覆盖。使用方法如下。

设置头信息:response.setHeader("头信息内容","头信息参数");

常用的刷新 refresh,例如:response.setHeader("refresh","1");

几秒后跳转:response.setHeader("refresh","2;URL=XXX")。

实战演练 2——网页保存为 Word 文档

【题目】

设计一个网页,要求的功能是能把当前网页保存为 Word 格式的文档。

【分析】

使用 response 对象的 setContentType 方法完成此题。

【运行效果】

打开开发工具 MyEclipse,在工程 jsp-nz 中的 WebRoot 文件夹下,新建工程文件 ex2.jsp。程序输入完毕并且保存之后,运行开发工具 MyEclipse 中的 Tomcat 服务器。最后在浏览器中输入地址 http://127.0.0.1:8080/jsp-nz/ex2.jsp 就可以看到运行结果了(如图 4-10～图 4-12 所示)。





图 4-10 ex2. jsp

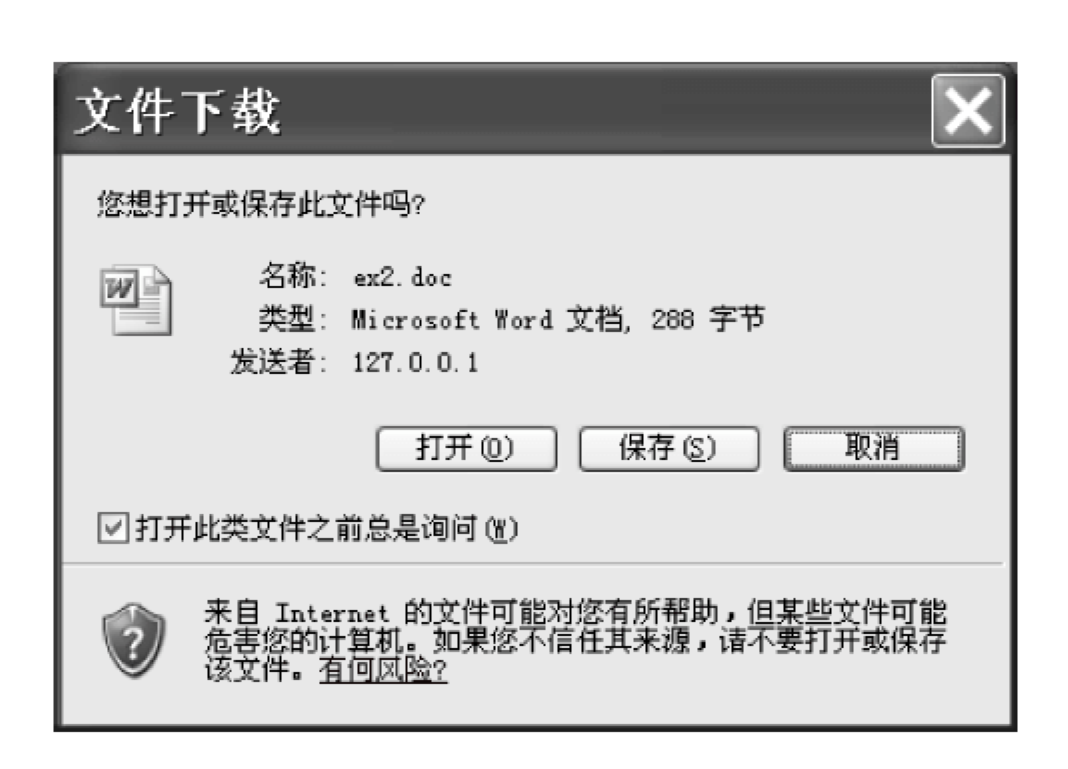


图 4-11 ex2. doc 下载

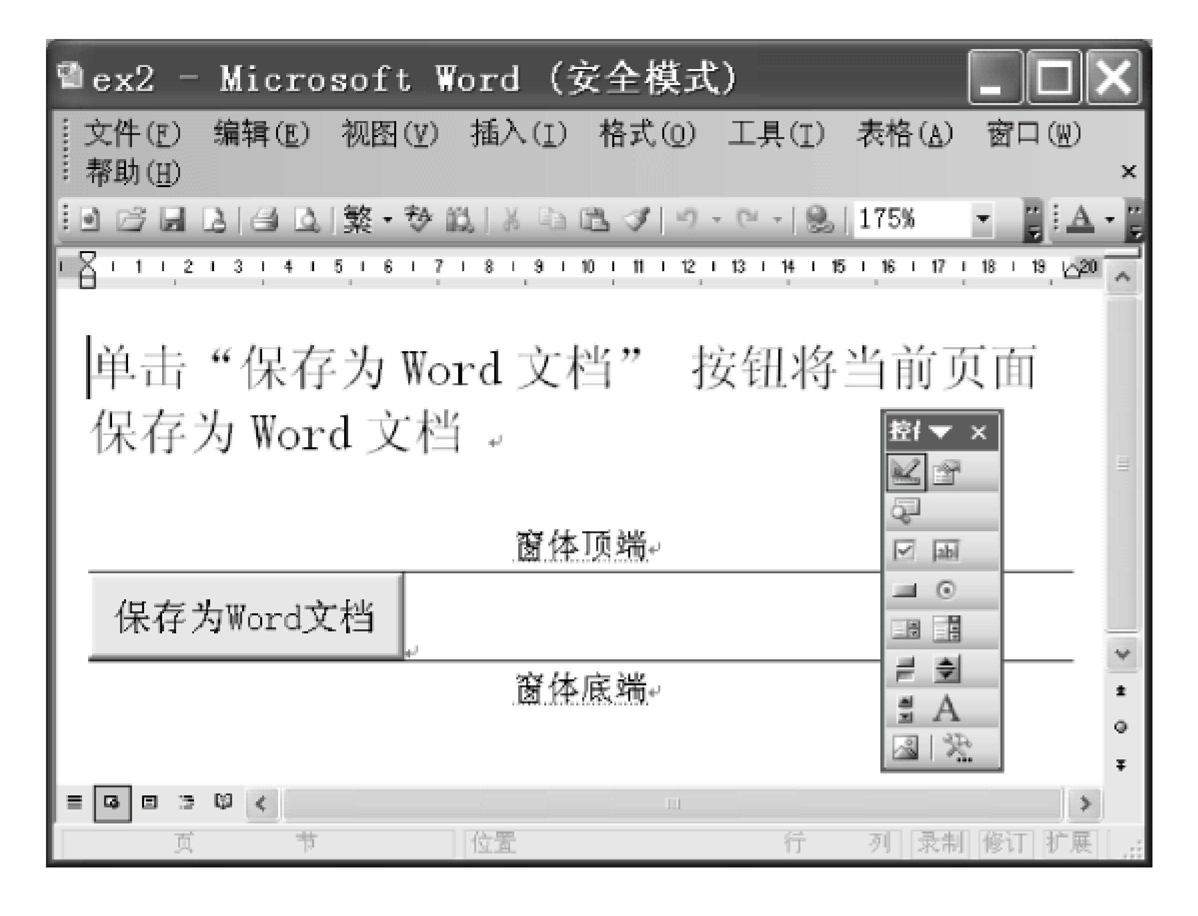


图 4-12 ex2. doc

【源代码】

【ex2. jsp】

```
<%@ page contentType="text/html;charset=gb2312" %>
<html>
 <head>
   <title>response 的 contentType 属性设置</title>
 </head>
 <body>
    <p>单击“保存为 Word 文档”按钮将当前页面保存为 Word 文档
      <form  action="" method="post" name="form" >
        <input type="submit"  name="word" value="保存为 Word 文档" >
    </form>
    <%
        String word=request.getParameter("word");
        if (word==null)
        {  word="";  }
        if(!word.equals(""))
        {response.setContentType("application/msword;charset=GB2312"); }
    %>
```

```
    </body>
</html>
```

实战演练 3——网页定时刷新

【题目】

设计一个网页，要求的功能是能把当前网页每隔 1 秒刷新 1 次。

【分析】

使用 response 对象的 setHeader 方法完成此题。

【运行效果】

打开开发工具 MyEclipse，在工程 jsp-nz 中的 WebRoot 文件夹下，新建工程文件 ex3.jsp。程序输入完毕并且保存之后，运行开发工具 MyEclipse 中的 Tomcat 服务器。最后在浏览器中输入地址 http://127.0.0.1:8080/jsp-nz/ex3.jsp 就可以看到运行结果了（如图 4-13 所示）。



图 4-13　ex3.jsp

【源代码】

【ex3.jsp】

```
<%@ page contentType="text/html;charset=gb2312" import="java.util.*"%>
<html>
  <head>
      <title>response 的 refresh 属性</title>
  </head>
  <body>
      <%
          Date date=new Date();
          response.setHeader("refresh","1");
      %>
      <p>页面将每隔 1 秒钟刷新一次。
      <p>现在的时间是   
      <%=date.getHours()%>:<%=date.getMinutes()%>:<%=date.getSeconds() %>
  </body>
</html>
```

4.3　session 对象

实战演练——简易购物

【题目】

这是一个简易的购物练习，在界面中输入顾客姓名和想要购买的商品，然后在购物车中显示出顾客姓名和已经购买的商品，实现这样一个虚拟购物的过程。

【分析】

题目需要三个文件来完成，ex4.jsp 用于输入顾客姓名；first.jsp 用于输入想要购买的商品；account.jsp 用于显示购物车中的商品和顾客的姓名。ex4.jsp 文件中需要一个表单提交顾客的姓名。first.jsp 文件中使用 session 对象的 set 方法存储顾客姓名，使用一个表单提交顾客想要购买的商品名称。account.jsp 文件中使用 session 对象的 get 方法获取顾客姓名，并且调用外部文件 hzsb.jsp 完成汉字识别的功能。

【运行效果】

打开开发工具 MyEclipse，在工程 jsp-nz 中的 WebRoot 文件夹下，新建工程文件 ex4.jsp、first.jsp、account.jsp。程序输入完毕并且保存之后，运行开发工具 MyEclipse 中的 Tomcat 服务器。最后在浏览器中输入地址 http://127.0.0.1:8080/jsp-nz/ex4.jsp 就可以看到运行结果了（如图 4-14～图 4-16 所示）。



图 4-14　ex4.jsp 购物登录首界面



图 4-15　first.jsp

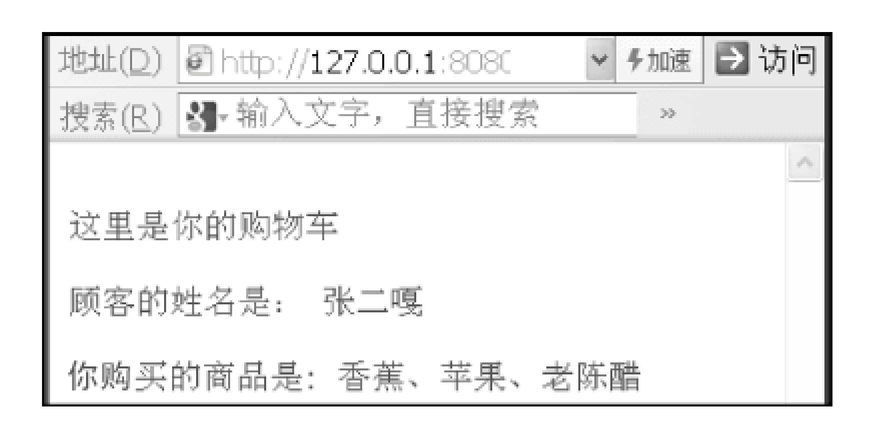


图 4-16　account.jsp

【源代码】

【ex4.jsp】

```
<%@ page contentType="text/html;charset=gb2312" language="java" %>
<html>
  <head><title>session 简易购物</title></head>
  <body>
    <font color="#aa2299" size=4>
       <p>输入你的姓名：
       <form action="first.jsp" methed="post" name="form">
           <input type="text" name="customer">
           <input type="submit" name="submit" value="去购物">
      </form>
    </font>
  </body>
</html>
```

【first. jsp】

```
<%@ page contentType="text/html;charset=gb2312" language="java" %>
<html>
  <body>
    <font color="#aa2299" size=4>
        <% String c=request.getParameter("customer");
            session.setAttribute("xm",c);
        %>
        <p>输入你想购买的商品:
          <form action="account.jsp" methed="post" name="form">
            <input type="text" name="goods">
            <input type="submit" name="submit" value="结账">
          </form>
    </font>
  </body>
</html>
```

【account. jsp】

```
<%@ page contentType="text/html;charset=gb2312" language="java" %>
<%@ include file="hzsb.jsp" %>
<html>
  <body>
    <font color="#aa2299" size=4>
      <% String name=(String)session.getAttribute("xm");
        name=hzsb(name);
        String g=hzsb(request.getParameter("goods"));
      %>
      <p>这里是你的购物车
      <p>顾客的姓名是:
      <%=name %>
      <p>你购买的商品是:
      <%=g%>
    </font>
  </body>
</html>
```

【hzsb. jsp】

```
<%@ page contentType="text/html;charset=gb2312" language="java" %>
<%!
    public String hzsb(String s)
    {
        String str=s;
        try
        {
            byte b[]=str.getBytes("ISO-8859-1");
            str=new String(b);
            return str;
        }
```

```
            catch(Exception e)
            {return str;}
        }
%>
```

【补充说明】

ex4.jsp 代码说明

第 7 行到第 10 行是一个表单，表单中有一个文本框和一个按钮，文本框用于输入顾客的姓名，变量名为 customer，表单将把顾客姓名提交到 first.jsp 文件。

first.jsp 代码说明

第 5 行到第 7 行是 Java 程序片，其中第 5 行声明一个字符串型变量 c，使用 request 对象的 get 方法接收 ex4.jsp 文件传递过来的顾客姓名；第 6 行使用 session 对象的 set 方法设置变量 xm 的值为 c 的值，本行代码使用 session 对象保存顾客姓名变量的值；第 9 行到第 12 行是一个表单，表单中有一个文本框和一个按钮，文本框用于输入想要购买的商品，变量名为 goods，表单将把所购商品提交到 account.jsp 文件。

account.jsp 代码说明

第 2 行，使用 include 指令引入 hzsb.jsp 文件，用于识别汉字；第 6 行到第 9 行是 Java 程序片，其中第 6 行使用 session 对象的 get 方法获取 xm 变量的值，可以获得顾客的姓名；第 7 行是对顾客的姓名进行汉字识别；第 8 行声明字符串变量 g 接收购物商品的名称，并且进行汉字识别；第 12 行输出顾客姓名；第 14 行输出商品名称。

4.3.1 session 对象

在开发 Web 应用程序时，开发者常会遇到这样的问题：当用户浏览网站时，如何跟踪和记录用户的一些特定信息(如用户身份)，而不用在每次向服务器发出请求时都让用户验证自己的身份。使用 session 对象就可以解决这个问题。当用户请求 Web 页面时，如果该用户还没有 session 对象，则 Web 服务器会为该用户创建一个属于自己的 session 对象，存储用户会话所需要的信息。这样，当用户在应用程序的各 Web 页之间跳转时，存储在 session 对象中的信息就会在整个会话过程中一直存在，直到会话过期或者被放弃，Web 服务器才会终止该会话。

注意：session 中保存的信息必须是对象类型，而不能是基本类型，如果声明为基本类型则需要转换。

4.3.2 session 对象常用方法

Session 对象的常用方法如下。

1. setAttribute()

此方法设定指定名字的属性，并将其存储在 session 对象中。调用方法如下：

```
setAttribute(String name,java.lang.Object value)
```

上面例题中的应用：

```
String c=request.getParameter("customer");
session.setAttribute("xm",c);
```

2. getAttribute()

此方法获得指定的 session 对象。其调用方法如下：

```
session.getAttribute(String name)
```

其中，参数 name 是指定 session 对象的名称。该方法返回一个 Object 对象，任何对象都可加入到 session 对象中，因此对返回的对象要用强制转换的方法，将得到的对象转换为此对象原来的类型。

上面例题中的应用：

```
String name=(String)session.getAttribute("xm");
```

4.4 application 对象

实战演练——使用 application 制作站点计数器

【题目】

制作一个站点访问计数器，要求不能通过刷新页面累计访问量。

【分析】

本题通过一个文件 ex5.jsp 即可实现。文件中声明一个 countPeople()方法用于累计网站的访问量，如果网站是首次被登录就为访问量 count 赋初始值，如果有新的用户访问网站就为访问量 count 累加。文件中还需要有一段程序片用来判断用户是否为新登录，如果是新登录就调用 countPeople()方法，完成访问量的累加。

【运行效果】

打开开发工具 MyEclipse，在工程 jsp-nz 中的 WebRoot 文件夹下，新建工程文件 ex5.jsp。程序输入完毕并且保存之后，运行开发工具 MyEclipse 中的 Tomcat 服务器。最后在浏览器中输入地址 http://127.0.0.1:8080/jsp-nz/ ex5.jsp 就可以看到运行结果了(如图 4-17 所示)。



图 4-17　ex5.jsp 站点计数器

【源代码】

【ex5.jsp】

```
<%@ page language="java" contentType="text/html;charset=gb2312"%>
<%!
synchronized void countPeople()
{
      ServletContext application=getServletContext();
```

```
6        Integer number=(Integer)application.getAttribute("Count");
7        if(number==null)
8        {   number=new Integer(1);
9            application.setAttribute("Count",number); }
10       else
11       {   number=new Integer(number.intValue()+1);
12           application.setAttribute("Count",number); }
13 }
14 %>
15 <html>
16   <head>
17     <title>不可刷新的访问计数器</title>
18   </head>
19   <body>
20     <%
21       Integer personnum=(Integer)application.getAttribute("Count");
22       if(session.isNew()|personnum==null)
23       {countPeople();}
24       Integer yournumber=(Integer)application.getAttribute("Count");
25     %>
26     <p>欢迎访问本站,您是第<%=yournumber %>个访问用户。
27   </body>
28 </html>
```

【补充说明】

ex5.jsp 代码说明

第 2 行到第 14 行是一个方法声明,第 5 行声明 ServletContext 类型的变量 application;第 6 行声明一个整形变量 number 并赋值为 Count;第 7 行是一个 if 语句,如果 number 为空,就说明网站是首次被访问;第 8 行是给 number 赋初值 1;第 9 行使用 application 对象的 set 方法给 Count 变量赋值为 number 的值;第 11 行是有新客户访问的情况下,给访问量进行累加;第 12 行是把更新之后的 number 值赋给 Count;第 20 行到第 25 行是一个 Java 程序片,第 21 行声明一个整形变量 personnum 并赋值为 Count;第 22 行是一个 if 条件语句,如果网站是首次被登录或者 personnum 的值为空,那么就调用 countPeople()方法;第 24 行代码声明一个整形变量 yournumber 并且赋值为 Count,如果没有新登录的人员 Count 值不变,否则 yournumber 将被赋值为新的 Count 值。

4.4.1 application 对象

application 对象能够为多个应用程序保存信息,对于一个 jsp 容器而言,各个用户共同使用一个 application 对象,这是与 session 对象的最大区别。服务器启动以后,将自动创建 application 对象,该对象能够一直保持,直到服务器关闭为止。

4.4.2 application对象常用方法

1. set Attribute()

该方法是为应用程序级变量赋值，若在 application 对象中还不存在这个变量，会添加这个变量。上面例题中的用法：

```
number=new Integer(1);
application.setAttribute("Count",number);
```

2. getAttribute()

此方法得到指定的 application 对象在数据对象的值。其调用方法如下：

```
application.getAttractive(string name)
```

上面例题中的用法：

```
Integer number=(Integer)application.getAttribute("Count");
```

其中，第一个对象 name 是指 application 对象中数据对象的名称，第二个参数 object 是这个对象对应的值，其类型是 Object，说明可以是任意类型，如果对象不存在，则添加该对象；如果已存在，则将其覆盖。

4.5 综合实战演练：购物车

【题目】

本题是一个购买图书的设计，需要在购物首界面显示图书的基本信息(书名、单价、出版社)，并且可以自己决定购买的数量，然后跳转到已购商品界面，确定所购商品准确无误之后可以结账，也可以在这时进行删除或再继续购买商品，在结算中心可以看到自己的购物清单合计。

【分析】

本题需要三个文件完成全部功能的实现，分别是购物文件 goods.html、已购商品文件 cart.jsp、结算中心文件 cost.jsp。

在 goods.html 文件中需要设计一个表格显示所有商品的三个基本信息，每个商品配备一个文本框和一个提交按钮以及三个隐藏组件用来记录商品的基本信息。

在 cart.jsp 文件中需要设计一个表单，表单中有一个表格，显示顾客购买并提交的商品信息，还需要给每个商品配备一个删除按钮，如果后悔了就可以不买，另外还有“继续购买商品”和“到结算中心”的链接。

在 cost.jsp 文件中，有一个表单，表单中有一个表格，显示顾客最终决定购买的商品，并且计算出所购商品的合计。

【运行效果】

打开开发工具 MyEclipse，在工程 jsp-nz 中的 WebRoot 文件夹下，新建工程文件 goods. html、cart. jsp、cost. jsp。程序输入完毕并且保存之后，运行开发工具 MyEclipse 中的 Tomcat 服务器。最后在浏览器中输入地址 http://127.0.0.1:8080/jsp-nz/ goods. html 就可以看到运行结果了(如图 4-18～图 4-20 所示)。



图 4-18　goods. html 购物界面



图 4-19　cart. jsp 已购商品界面



图 4-20　cost. jsp 结算界面

【源代码】

【goods. html】

```
<html>
  <head>
    <title>购物车实例</title>
```

```
  </head>
  <body>
    <h1 align="center" >购  物  车</h1>
    <table width="90%" border="0" align="center" cellpadding="10" cellspacing="1"
    bgcolor="#CC9900">
      <tr>
         <td width="30%"> 书名</td>
         <td width="10%">单价<br></td>
         <td width="30%">出版社<br></td>
         <td width="30%">购买数量<br></td>
      </tr>
      <tr>
         <td bgcolor="#FFFFCC"> 安徒生童话</td>
         <td bgcolor="#FFFFCC">45.00<br></td>
         <td bgcolor="#FFFFCC">教育出版社<br></td>
         <td bgcolor="#FFFFCC">
          <form method="post" action="cart.jsp">
             <input type="text" name="num" value="1" size="3">
             <input type="submit" value="放到购物车" name="button6">
             <input type="hidden" value="安徒生童话" name="sm">
             <input type="hidden" value="45.00" name="cost">
             <input type="hidden" value="教育出版社" name="cbs">
          </form>
        </td>
      </tr>
      <tr>
         <td bgcolor="#FFFFCC"> 格林童话</td>
         <td bgcolor="#FFFFCC">30.00<br></td>
         <td bgcolor="#FFFFCC">教育出版社<br></td>
         <td bgcolor="#FFFFCC">
          <form method="post" action="cart.jsp">
             <input type="text" name="num" value="1" size="3">
             <input type="submit" value="放到购物车" name="button7">
             <input type="hidden" value="格林童话" name="sm">
             <input type="hidden" value="30.00" name="cost">
             <input type="hidden" value="教育出版社" name="cbs">
          </form>
        </td>
      </tr>
    </table>
  </body>
</html>
```

【cart.jsp】

```
<%@ page language="java" contentType="text/html;charset=gb2312"%>
<%@ page import="java.util.*" %>
<%@ include file="hzsb.jsp" %>
<html>
  <head>
```

```
    <title>我的购物车</title>
  </head>
  <body>
    <h1 align="center" >已  购  商  品</h1>
    <% String[] goods=new String[4];
      goods[0]=request.getParameter("sm");
      goods[1]=request.getParameter("cost");
      goods[2]=request.getParameter("cbs");
      goods[3]=request.getParameter("num");
      Vector goodslist=(Vector)session.getAttribute("goodscart");
      if(goodslist==null)
      {  goodslist=new Vector();
         goodslist.addElement(goods);
      }
      else
      {  goodslist.addElement(goods);
         if(request.getParameter("delgoods")!=null)
         {  String delgoods=request.getParameter("delgoods");
            goodslist.removeElementAt(Integer.parseInt(delgoods));
         }
      }
      session.setAttribute("goodscart",goodslist);
    %>
    <table width="90%" border="0" align="center" cellpadding="10" cellspacing="1"
    bgcolor="#CC9900">
      <tbody><tr>
        <td width="30%"> 书名</td>
        <td width="10%">单价  </td>
        <td width="30%">出版社  </td>
        <td width="10%">购买  </td>
        <td width="20%">删除  </td>
    </tr>
    <% if(goodslist!=null)
      for(int x=0;x<goodslist.size();x++)
      {
        String[] buygoods;
        buygoods=(String[])goodslist.elementAt(x);
        if(buygoods[0]!=null)
        {String sm=hzsb(buygoods[0]);
        String cbs=hzsb(buygoods[2]);
    %>
    <tr>
      <td bgcolor="#FFFFCC"> <%=sm%></td>
      <td bgcolor="#FFFFCC"> <%=buygoods[1]%></td>
      <td bgcolor="#FFFFCC"> <%=cbs%></td>
      <td bgcolor="#FFFFCC"> <%=buygoods[3]%></td>
      <td align="center" valign="middle" bgcolor="#FFFFCC">
        <form method="post"> 
          <input type="submit" value="从购物车中删除" name="button8">
```

```
            <input type="hidden" value="del" name="del">
            <input type="hidden" value=<%=x %> name="delgoods">
          </form></td></tr>
      <%}} %>
        </tbody></table>
      <h2 align="center"><a href="goods.html"> 继续购买商品</a>
           <a href="cost.jsp">到结算中心</a>
      </h2>
    </body>
</html>
```

【cost.jsp】

```
<%@ page language="java"
contentType="text/html;charset=gb2312" %>
<%@ page import="java.util.*" %>
<%@ include file="hzsb.jsp" %>
<html>
<head>
  <title>已购商品明细</title>
</head>
<body><h1 align="center" >结  算  中  心</h1>
  <table width="90%" height="113" border="0" align="center" cellpadding="10"
  cellspacing="1" bgcolor="#CC9900">
    <tr>
      <td width="30%"> 书名</td>
      <td width="10%">单价  </td>
      <td width="30%">出版社  </td>
      <td width="30%">购买  </td>
    </tr>
    <%
      Vector goodslist=(Vector)session.getAttribute("goodscart");
      float totalmoney=0;
      for(int x=0;x<goodslist.size();x++)
      {  String[] buygoods;
         buygoods=(String[])goodslist.elementAt(x);
         if(buygoods[0]!=null)
         {String sm=hzsb(buygoods[0]);
         String cbs=hzsb(buygoods[2]);
    %>
    <tr>
      <td bgcolor="#FFFFCC"> <%=sm %></td>
      <td bgcolor="#FFFFCC"> <%=buygoods[1] %></td>
      <td bgcolor="#FFFFCC"> <%=cbs %></td>
      <td bgcolor="#FFFFCC"> <%=buygoods[3] %></td></tr>
    <% totalmoney+=((Float.parseFloat(buygoods[1]))*(Float.parseFloat(buygoods[3])));
      }     }
    goodslist.removeAllElements();
    %>
```

```
35      <tr>
36        <td bgcolor="#FFFFCC"> <a href="goods.html">返回购物首页</a>
          </td>
37        <td bgcolor="#FFFFCC"> </td>
38        <td bgcolor="#FFFFCC"> </td>
39        <td bgcolor="#FFFFCC">合计  <%=totalmoney %></td></tr>
40      </tbody></table>
41    </body>
42  </html>
```

【补充说明】

goods.html 补充说明

第 7 行到第 42 行，是一个完整的表格；第 8 行到第 13 行，是表格的第一行，这行有四个单元格；第 14 行到第 27 行，是表格的第二行，这行前三个单元格是商品的基本信息，第四个单元格是一个表单，表单信息将要提交到 cart.jsp 文件，表单中包含一个文本框，变量名 num，赋初值为 1，一个提交按钮，三个隐藏按钮用于存放商品的三个基本信息，变量名分别为 sm(书名)、cost(单价)、cbs(出版社)；第 28 行到第 41 行，是表格的第三行，这行放置第二个商品。

cart.jsp 补充说明

第 2 行引入一个 util 类包；第 3 行引入汉字识别文件 hzsb.jsp；第 10 行声明一个字符串类型的数组，数组名称为 goods，数组总共有 4 个成员；第 11 行到第 14 行使用数组的四个成员分别接收了商品的书名 sm、单价 cost、出版社 cbs、购买数量 num，这些信息都是由 goods.html 文件传递过来的；第 15 行使用 Vector 类型定义了一个向量 goodslist，用于存放商品信息；第 16 行是一个 if 语句，判断向量 goodslist 是否为空，如果为空，就表示顾客是首次购买；第 17 行到第 19 行，如果是首次购买的情况，就为向量 goodslist 申请一个新的空间，然后将数组 goods 中的所有成员信息都放到 goodslist 的向量列表中；第 21 行，如果不是首次购买，就直接将数组 goods 中的所有成员信息都添加到 goodslist 的向量列表中；第 22 到第 25 行，是一组 if 语句，判断条件是接收 delgoods 变量的值，判断是否有需要从购买清单中删除的商品，如果 delgoods 变量的值不为空，就从 goodslist 的向量列表中删除相对应的商品；第 27 行，设置 goodscart 变量的值为 goodslist 向量列表的值；第 29 行到第 58 行是一个表格，第 30 行到第 36 行是表格第一行标题行；第 37 行到第 57 行是一组 if 判断语句嵌套 for 循环语句，首先判断如果 goodslist 向量不为空，那么就把向量列表中所有的商品信息都依次显示到表格下一行相对应的各个单元格中，并且在每一个商品后面都对应一个表单，表单中有一个删除按钮和一个隐藏按钮，隐藏按钮用于标记是否删除该商品，变量名为 delgoods，删除按钮用于执行表单的信息提交，默认提交到本文件中；第 59 行是继续购置商品的链接 goods.html；第 60 行是到结算中心的链接 cost.jsp。

cost.jsp 补充说明

第 2 行引入一个 util 类包；第 3 行引入汉字识别文件 hzsb.jsp；第 9 行到第 40 行是一个表格；第 17 行声明一个 Vector 类型的向量 goodslist 接收 goodscart 的值；第 19 行

到第32行是一个for循环语句，用于提取向量goodslist列表中各个商品的信息显示在表格的相应单元格中，并且计算商品价格的累加得到所有商品的合计；第33行移除向量goodslist中的所有商品列表；第36行提供了返回购物首页的链接goods.html。

本章小结

本章讲述了JSP内置对象的应用，重点讲述了request、response、session、application、out这五个常用的对象。

JSP内置对象包括request、response、session、application、out、config、pageContext，这些对象在服务器和客户端交互的过程中分别完成不同的功能。

1. Response

Response对象与HttpServletResponse类关联，可用来向客户端输入数据。

2. Session

Session对象与HttpSession类关联，可用来保存服务区端与一个客户端之间需要保留的数据，当客户端关闭网站（或称为系统）的所有网页时，session变量会自动清除。HTTP是一个无状态协议，不保留会话间的数据，可通过session对象扩展HTTP的功能。如果使用page指令关闭了session，则在此页面中调用session对象时将导致错误，因此在工程实践中，一般不必关闭session。

Request、response、session是JSP内置对象中3个重要的对象，这3个对象提供了服务器端与客户端（即浏览器）进行交互通信的控制。当客户端打开浏览器，在地址栏中输入服务器Web服务页的地址时，是使用HTTP协议，向服务器发送了一个请求，服务器在收到来自客户端发送的请求后，通过response对客户端进行响应，而session就会保存这个会话期间需要使用的数据信息。

请求的方法有多种，如post，get，put，head，delete，trace等，最常用的是前两种。Request使response对象还可以操作HTTP头中的数据。

3. Application

Application对象与ServletContext类关联，一旦创建（在服务器开始提供服务器时，即第一个被访问时application对象就会被创建），就会一直保留到服务器关闭服务为止。所以application对象可用来提供一些全局的数据和对象。

本章习题

1. 教师在登录界面输入用户名和密码之后可以进入教案编辑页面，填写完毕之后可以提交教案内容。在整个过程中，从教师登录之后，用户名就一直显示在页面上，并且教

案内容可以以中文的形式填写和提交。

【分析】

题目总共需要四个文件来完成，分别命名为 denglu-t. jsp、plan. jsp、plan-1. jsp、hzsb. jsp。

denglu-t. jsp 文件是登录界面，填写完登录信息之后提交到 plan. jsp 页面，这个题目中不包含判断用户名和密码是否输入正确的功能。

在 plan. jsp 文件使用 session 对象的 set 属性记录用户名的信息，从这个页面开始，后面的页面想要显示用户名信息的地方都可以通过 session 对象的 get 属性获取用户名。这个页面也是制作教案的页面，教案内容填好之后，提交到 plan-1. jsp。

在 plan-1. jsp 文件接收教案的全部信息，并且使用 include 指令引入 hzsb. jsp 文件，在需要识别汉字的地方调用 hzsb()方法来识别教案中输入的全部中文，以便显示在网页上。

【运行】

打开开发工具 MyEclipse，新建工程 plan，并在 plan 的 WebRoot 文件夹下新建工程文件 denglu-t. jsp、plan. jsp、plan-1. jsp、hzsb. jsp。程序输入完毕并且保存之后，运行 Tomcat 服务器。最后在浏览器中输入地址 http://127. 0. 0. 1:8080/plan/denglu-t. jsp 就可以看到运行结果了（如图 4-21～图 4-23 所示）。

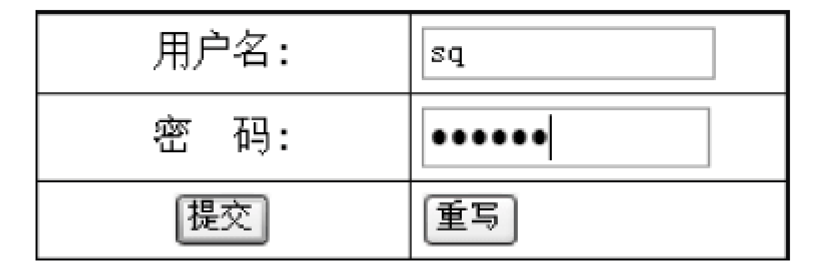


图 4-21　登录界面 denglu-t. jsp

			sq老师您已登录
授课教师		日期	
授课班级		课时	
授课内容			
	提交		

图 4-22　教案编辑页面 plan. jsp

			sq老师您已登录
	教案		
授课教师	孙茜	日期	2014-9-2
授课班级	12网络班	课时	2
授课内容	JSP内置对象：session保存用户信息		

图 4-23　教案提交页面 plan-1. jsp

2. 本题是在上一题目的基础上的拓展，这是一个分权限登录的系统，需要输入正确的用户名和密码，并且选择权限之后才能登录，如果是学生登录，可以做测验题，在测验界

面可以显示当前的时间和用户名,并且可以保存试题和得出测验分数。

如果是教师登录,登录之后可以编辑教案,在这个页面可以显示当前的时间和用户名,教案写好之后可以完成提交。

【分析】

denglu.jsp:登录首页面。

teacher.jsp:设置了教师登录正确的用户名的密码。使用 session 对象的 set 属性记录了教师的登录信息,后面的页面想要显示教师信息的地方都可以通过 session 对象的 get 属性获取。

student.jsp:设置了学生正确登录的用户名的密码。使用 session 对象的 set 属性记录了学生的登录信息,后面的页面想要显示学生信息的地方都可以通过 session 对象的 get 属性获取。

denglu-1.jsp:设置了用户通过登录信息检测后具体登录到哪个二级页面。

time.jsp:能够显示当前时间。

test.jsp:学生登录之后进入这个测验页面,在页面的上方显示了学生登录用户名,通过 session 对象的 get 属性获取。页面还能显示当前时间,通过 include 指令引入 time.jsp 实现。

test-1.jsp:学生做过测试之后可以把题目提交,这个页面用于计算测验成绩,并把最终成绩显示在 test.jsp 页面。

plan.jsp:教师登录后可以做教案编辑,这个页面显示了教师登录信息和当前时间。

plan-1.jsp:教案编辑完成后提交到此页面,在这个页面完成显示教案全部内容,使用 request 对象的 getParameter 方法获取表单提交的值,并且使用汉字识别方法来完成显示。

hzsb.jsp:完成汉字识别功能。

【运行】

打开开发工具 MyEclipse,新建工程 tz,并在 tz 的 WebRoot 文件夹下新建工程文件 denglu.jsp、denglu-1.jsp、teacher.jsp、student.jsp、time.jsp、test.jsp、test-1.jsp、plan.jsp、plan-1.jsp、hzsb.jsp。程序输入完毕并且保存之后,运行 Tomcat 服务器。最后在浏览器中输入地址 http://127.0.0.1:8080/tz/denglu.jsp 就可以看到运行结果了(如图 4-24～图 4-32 所示)。

用户名:	ss
密　码:	••
◉学 生	○教 师
提交	重写

图 4-24　学生登录

这是学生登录页面!登录失败!!

返回登录首页

图 4-25　学生登录失败

这是学生登录页面!

测验

返回登录首页

图 4-26　学生登录成功

ss学生已登录， 当前时间:2014-8-18 17:15:09 保存试题

测 验

1、字符串连接符号是（ ）

○ A)+ ○ B)& ○ C)&& ○ D)-

2、（ ）用来设置整个JSP页面的相关属性及功能

○ A)page指令 ○ B)include指令 ○ C)taglib指令 ○ D)以上都是

3、JSP页面能够通过（ ）来包含其他JSP文件、HTML文件或文本文件

○ A)page指令 ○ B)include指令 ○ C)taglib指令 ○ D)以上都是

4、搭建JSP环境需要下面选项中的哪些（ ）

□ A)Tomcat5.5/Tomcat6.0 □ B)JDK1.6 □ C)Dreamweaver □ D)浏览器

提交答案 重新作答 本次测验结果2分

图 4-27 学生已登录

用户名：	sun
密　码：	••
○学 生	◉教 师
提交	重写

图 4-28 教师登录

这是教师登录页面！登录失败！！

返回登录首页

图 4-29 教师登录失败

这是教师登录页面！

写教案

返回登录首页

图 4-30 教师登录成功



图 4-31 教师登录后，编辑教案

sun老师已登录， 当前时间:2014-8-18 17:20:18

教 案			
授课教师	孙茜	日期	2014-9-2
授课班级	13网络两年制	课时	2课时
授课内容	综合训练		

安全退出

图 4-32 教案提交成功

情境五

JSP 数据库应用

Project 5

【学习目标】

1. 了解数据库的基本概念。
2. 掌握 SQL 语句。
3. 掌握 JDBC-ODBC 技术。
4. 能够使用 JDBC-ODBC 技术访问数据库。

5.1 SQL 数据库的安装

5.1.1 SQL Server 2000 的安装

(1) 运行 SQL Server 2000 的安装程序,弹出如图 5-1 所示的安装界面,单击“安装 SQL Server 2000 简体中文个人版”选项,弹出如图 5-2 所示的对话框,单击“安装 SQL Server 2000 组件”选项,弹出如图 5-3 所示的对话框,单击“安装数据库服务器”选项,弹出如图 5-4 所示的欢迎对话框。



图 5-1　安装界面 1



图 5-2　安装界面 2



图 5-3　安装界面 3

(2) 在欢迎对话框单击“下一步”,弹出如图 5-5 所示的计算机名对话框,这个对话框主要是用来设置创建 SQL Server 的计算机名称,在这里采用默认设置,单击“下一步”,弹出“安装选择”对话框,如图 5-6 所示。

(3)“安装选择”对话框主要供用户选择安装的方式,在这里采用默认设置,单击“下一步”进入“用户信息”对话框,如图 5-7 所示。



图 5-4 “欢迎”对话框



图 5-5 “计算机名”对话框



图 5-6 “安装选择”对话框



图 5-7 “用户信息”对话框

(4)“用户信息”对话框主要用于用户输入自己的信息,输入完毕单击“下一步”,进入“安装定义”对话框,这个对话框主要用于供用户选择安装类型,在这里选择“服务器和客户端工具”选项,如图 5-8 所示,单击“下一步”,进入“实例名”对话框,如图 5-9 所示。



图 5-8 “安装定义”对话框

(5)在“实例名”对话框中,用户可以选择安装或维护的 SQL Server 命名实例,在这里采用默认设置,单击“下一步”进入“安装类型”对话框,如图 5-10 所示。

(6)“安装类型”对话框中有 3 种安装类型供用户选择:典型、最小和自定义,还可以自定义程序文件和数据库文件的安装路径,在这里采用默认设置,单击“下一步”进入“服务账户”对话框,如图 5-11 所示。

(7)“服务账户”对话框主要用于设置服务设置,在这里采用默认设置,然后中服务设置中选择“使用本地系统账户”,单击“下一步”进入“身份验证模式”对话框,如图 5-12 所示。

(8)“身份验证模式”对话框主要用于设置 SQL Server 2000 的登录,在这里采用默认设置,单击“下一步”,直到最后程序安装完毕。



图 5-9　"实例名"对话框



图 5-10　"安装类型"对话框



图 5-11　"服务账户"对话框

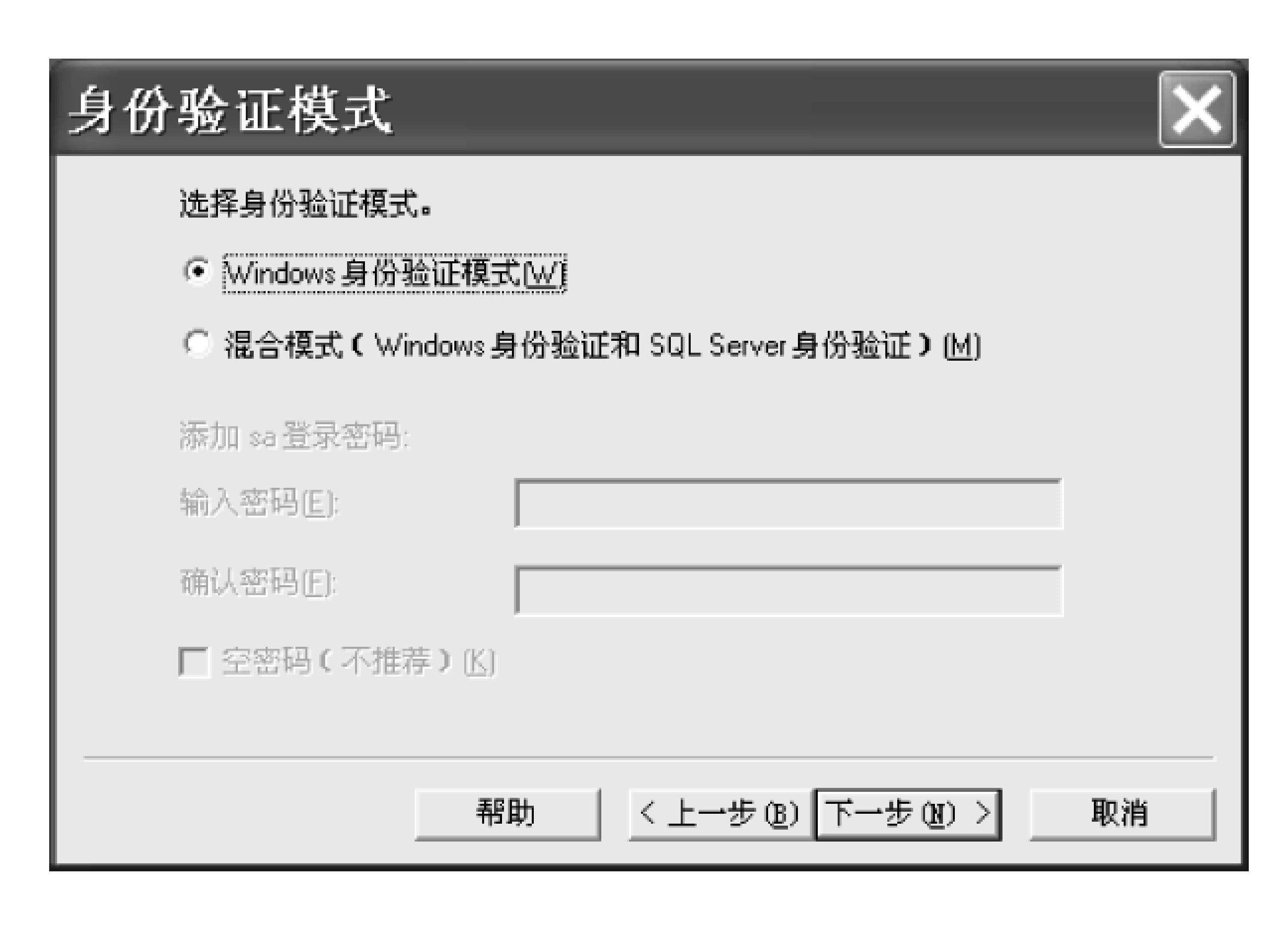


图 5-12 身份验证

5.1.2 SQL Server 2000 的使用

（1）启动 SQL Server 服务管理器。单击 Windows 桌面任务栏中的“开始”按钮，在开始菜单选择“程序”—“Microsoft SQL Server”—“服务管理器”，如图 5-13 所示，打开服务管理器，使用默认的服务器和服务，单击“开始/继续”按钮，如图 5-14 表示服务器正在运行。



图 5-13 启动服务管理器



图 5-14　从“程序”启动服务管理器

（2）启动 SQL Server 企业管理器。单击 Windows 桌面任务栏中的“开始”按钮，在开始菜单选择“程序”—“Microsoft SQL Server”—“企业管理器”，如图 5-15 所示，打开企业管理器如图 5-16 所示，该工具提供了图形化用户界面，以方便用户进行各种数据库操作。

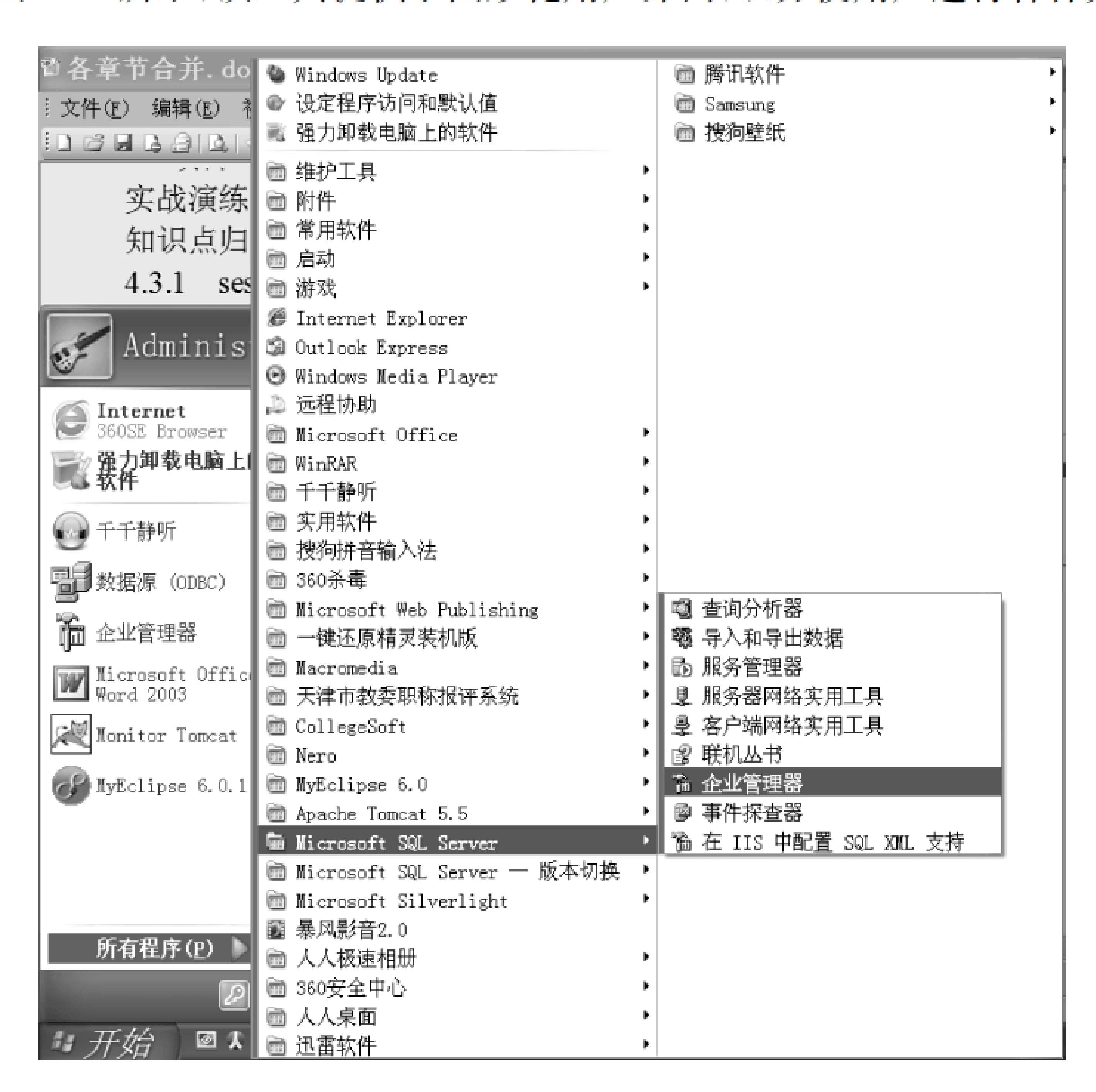


图 5-15　企业管理器

（3）备份数据库。当我们为工程新建了一个数据库并且想要把数据库保存在该工程的文件夹中的时候，就需要使用数据库的备份功能。首先在需要备份的数据库上单击鼠标右键，弹出快捷菜单选择“所有任务”—“备份数据库”，如图 5-17 所示，弹出备份对话框如图 5-18 所示，数据库名默认，备份选项选择“数据库—完全”，然后单击“添加”按钮打开“选择备份目的”对话框如图 5-19 所示。



图 5-16 企业管理器

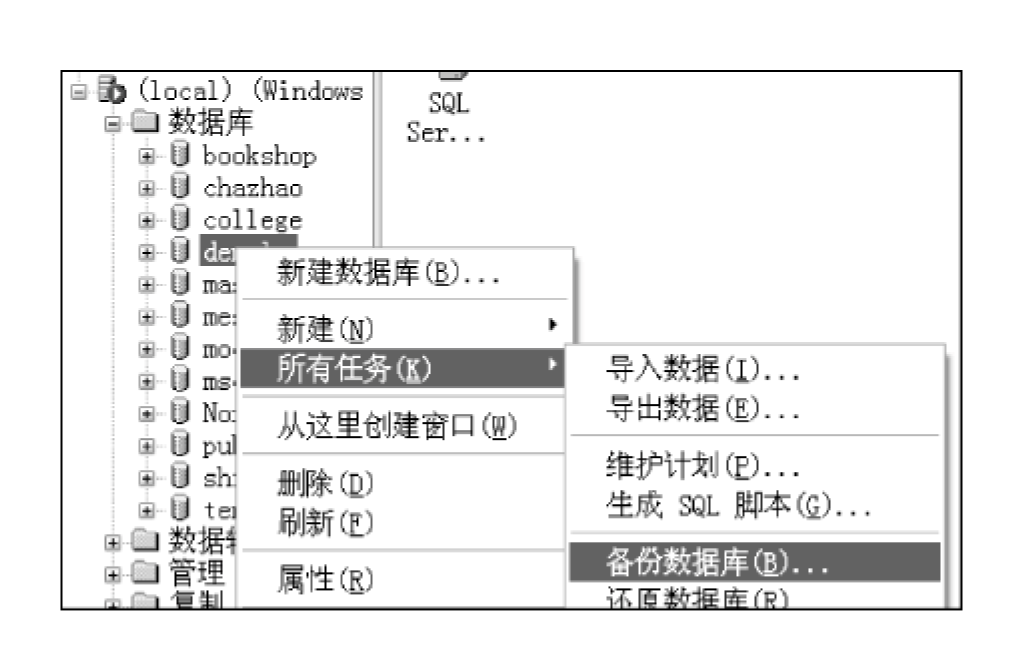


图 5-17 备份数据库

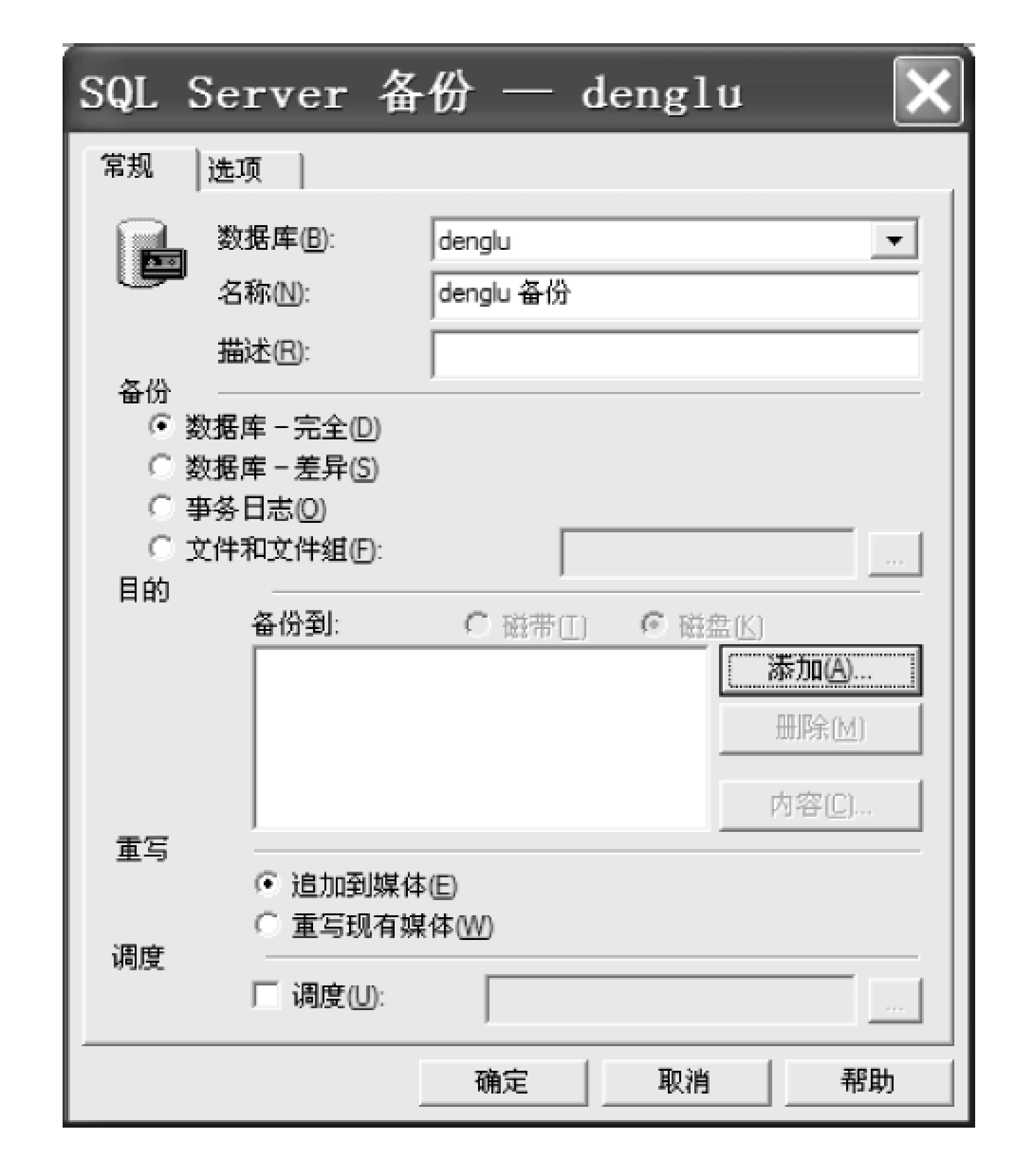


图 5-18 “SQL Server 备份”对话框



图 5-19 “选择备份目的”对话框

在“选择备份目的”对话框中，选择“文件名”选项，并单击“…”按钮，弹出“备份设备位置”对话框，如图 5-20 所示，选定好想要保存数据库的位置，并给数据库起好文件名，一般我们使用数据库本来的名字命名，如图 5-20 所示，最后单击“确定”按钮完成数据库的备份，弹出完成对话框如图 5-21 所示。

(4) 还原数据库，这个功能也是我们后面经常会用到的，当想要把外部数据库复制到企业管理器中进行操作时，还原数据库这个功能就可以完成这项操作了。首先需要新建一个与要还原的数据库名称一样的空数据库，然后在这个空的数据库上单击鼠标右键，弹出快捷菜单，选择“所有任务”—“还原数据库”，如图 5-22 所示，弹出“还原数据库”对话框如图 5-23 所示，数据库名已默认，还原项上选择“从设备”选项，备份选项选择“数据库—完全”，然后单击“选择设备”按钮打开“选择还原设备”对话框如图 5-24 所示。



图 5-20　“备份设备位置”对话框



图 5-21　“备份完成”对话框　　　　图 5-22　还原数据库



图 5-23　“还原数据库”对话框



图 5-24 “选择还原设备”对话框

在“选择还原设备”对话框上单击“添加”按钮，弹出“选择还原目的”对话框如图 5-25 所示，在这个对话框中单击“…”按钮，弹出“备份设备位置”对话框，如图 5-26 所示，选择要还原的数据库，然后单击“确定”按钮，直到最后完成还原，如图 5-27 所示。



图 5-25 “选择还原目的”对话框



图 5-26 “备份设备位置”对话框



图 5-27 “还原成功”对话框

5.2　使用 JDBC—ODBC 技术连接数据库

实战演练——用户的注册登录

【题目】

题目要求使用数据库技术，新建数据库名称为 denglu，会员表为 customer。完成的功能为，用户能够在登录界面注册成为网站的会员，并且能够作为会员登录网站。

【分析】

要完成本题目需要先理清思路一步一步地完成，步骤如下。

(1) 本题要完成的功能模块如图 5-28 所示。

(2) 打开 SQL 软件的企业管理器，新建数据库 denglu，如图 5-29 和图 5-30 所示。

(3) 在 denglu 数据库中新建会员用户表 customer，表的设计情况如图 5-31 所示。

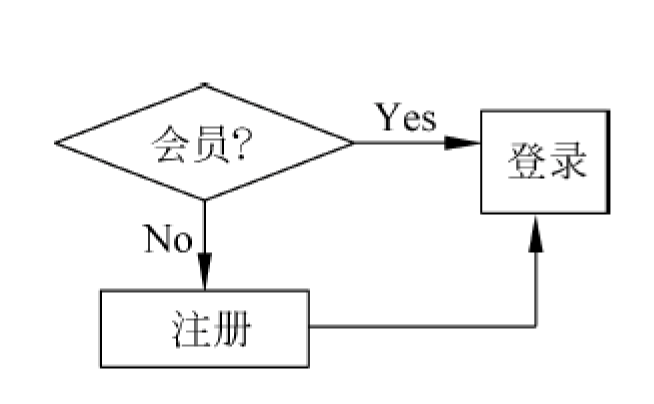


图 5-28　登录注册模块



图 5-29　新建数据库



图 5-30　新建数据库 denglu

列名	数据类型	长度	允许空	备注
id	varchar	10		用户名
password	varchar	15		密码
name	varchar	15		姓名
sex	varchar	8		性别
address	varchar	50		住址
code	varchar	15		邮编
tel	varchar	20		电话
email	varchar	30		邮箱

图 5-31　customer 表

(4) 建立 ODBC 数据源，打开系统管理工具——ODBC 数据源管理器对话框，单击“添加”按钮(如图 5-32 和图 5-33 所示)，弹出新建新数据源对话框(如图 5-34 所示)，选择最后一项 SQL Server，然后单击“完成”按钮。

然后弹出“创建到 SQL Server 的新数据源”对话框(如图 5-35 所示)，数据源名称命名为数据库的名称 denglu，服务器项选择本地服务器的名称。单击“下一步”弹出对话框(如图 5-36 所示)，再单击“下一步”弹出对话框(如图 5-37 所示)，选择默认数据库为 denglu。



图 5-32 控制面板——管理工具



图 5-33 “ODBC 数据源管理器”对话框



图 5-34 创建 SQL 新数据源

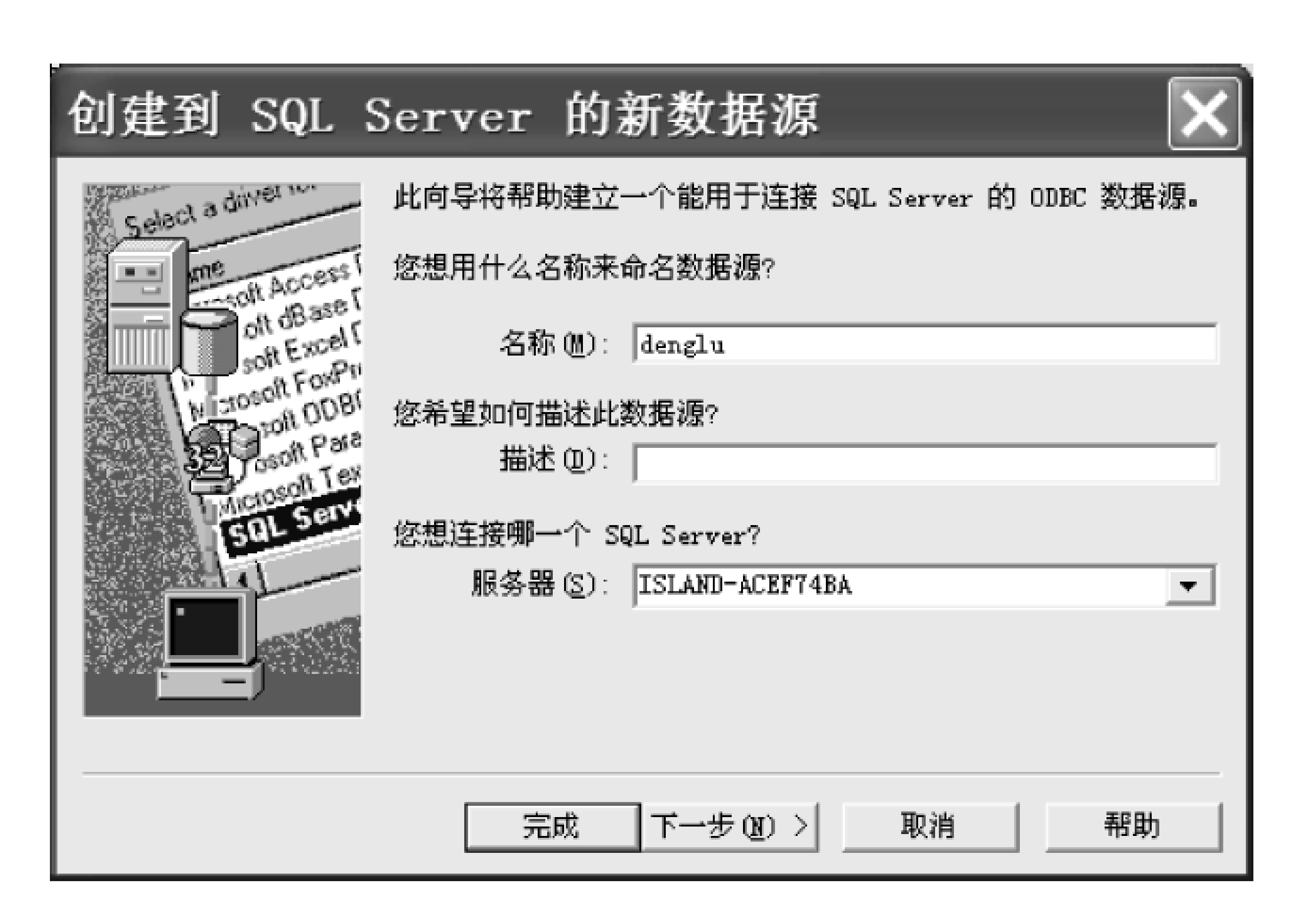


图 5-35　数据源名称 denglu

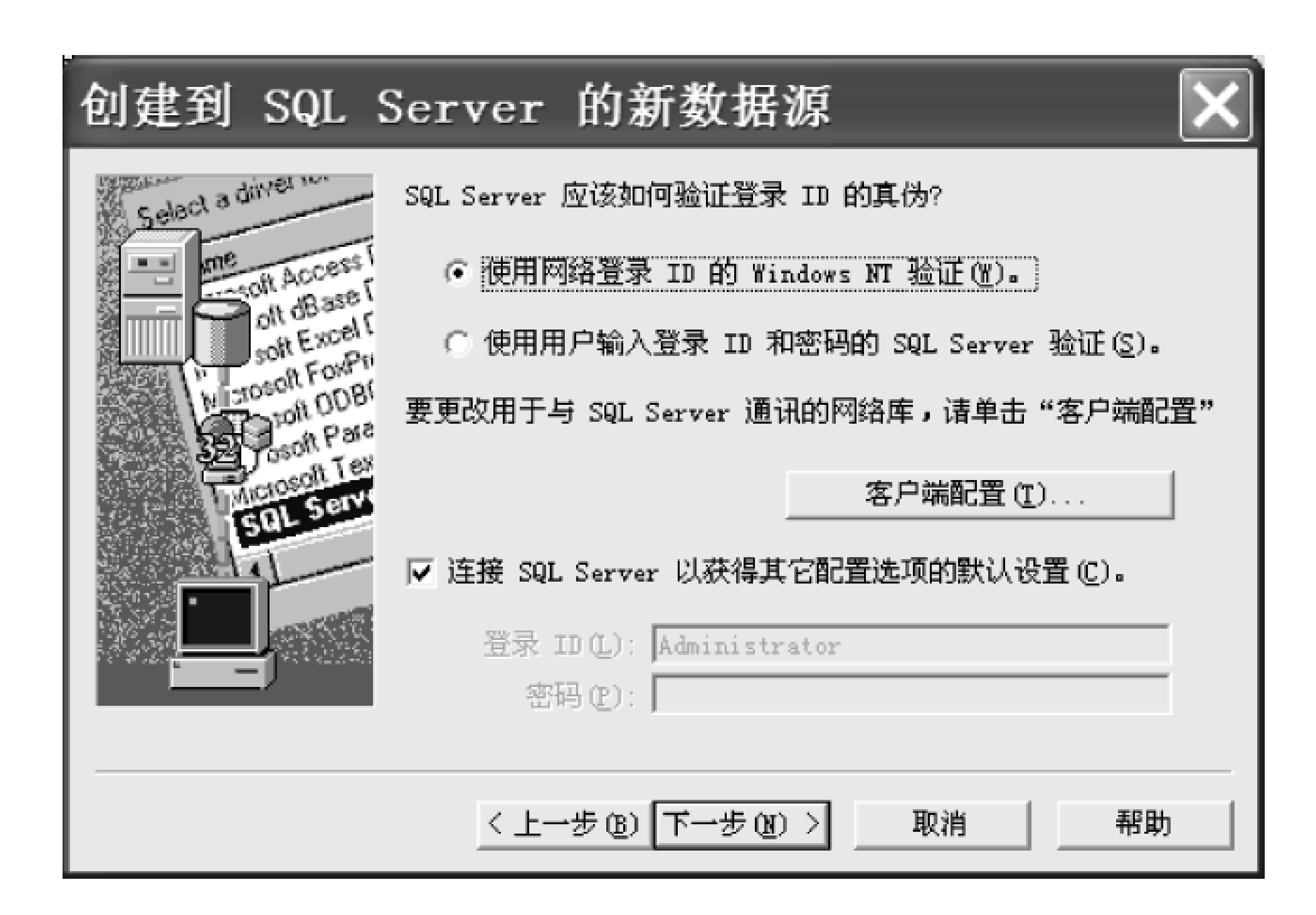


图 5-36　SQL Server 验证



图 5-37　选择默认数据库 denglu

然后再单击“下一步”，弹出对话框（如图 5-38 所示），最后单击“完成”按钮就可以看到已经添加好的数据源了（如图 5-39 所示）。



图 5-38 新数据源设置



图 5-39 denglu 数据源成功添加

（5）题目中需要的代码文件有：mydb.java 完成保存连接数据库的代码和用于对数据库进行增删改查以及汉字识别的方法；denglu.html 完成登录界面的设计；denglu.jsp 完成查找数据库中的 customer 表中的会员信息；zc.html 完成会员注册的界面设计；zc_1.jsp 完成会员信息添加到数据库的 customer 表；success.jsp 完成登录成功信息的显示。

【运行效果】

打开开发工具 MyEclipse，新建一个工程 jsp-jdbc，在 src 文件夹下新建 bean 文件夹，在 bean 文件夹中新建 mydb.java 文件；在这个工程的 WebRoot 文件夹下，新建文件 denglu.html、denglu.jsp、zc.html、zc_1.jsp、success.jsp。程序输入完毕并且保存之后，运行开发工具 MyEclipse 中的 Tomcat 服务器。最后在浏览器中输入地址 http://

127.0.0.1:8080/jsp-jdbc/denglu.html 就可以看到运行结果了(如图 5-40～图 5-43 所示)。

图 5-40　登录界面 denglu.html

图 5-41　注册界面 zc.html

图 5-42　会员登录界面 denglu.html

图 5-43　登录成功 success.jsp

【源代码】

【mydb.java】

```
package bean;
import java.sql.*;
public class mydb {
  private Connection conn;
  private Statement stmt;
  String message;
  public void connection()
  {   try
       { Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
        conn=DriverManager.getConnection("jdbc:odbc:denglu");
        stmt=conn.createStatement();}
      catch(ClassNotFoundException e)
       { message="connection"+e;}
       catch(SQLException e)
       { message="query"+e;}
  }
  public ResultSet query(String sql)
  {   ResultSet rs=null;
       try
       { rs=stmt.executeQuery(sql);}
       catch(SQLException e)
       {   message="query"+e;}
       return rs;
  }
  public int update(String sql)
  {  int n=0;
     try
     {   n=stmt.executeUpdate(sql);}
     catch(SQLException e)
     {   message="update"+e;}
     return n;
  }
  public boolean delete(String sql)
  {  boolean b=false;
     try
         { b=stmt.execute(sql);}
     catch(SQLException e)
         {   message="delete"+e;}
     return b;
  }
  public String hzsb(String s)
  {   String str=s;
       try
          {byte b[]=str.getBytes("ISO-8859-1");
            str=new String(b);
            return str; }
```

```
47        catch(Exception e)
48            {return str;}
49    }
50 }
```

mydb.java 补充说明

第 1 行,说明 mydb.java 是在 bean 文件夹中的。

第 2 行,引入 java 的 SQL 包,因为在下面的代码中有关于 SQL 数据库的命令操作。

第 3 行,声明一个公共类,类的名字为 mydb,与这个文件的文件名相同,是在文件生成的时候自动生成的类名。

第 4 行,conn 变量是公共类的私有变量,用于完成数据库连接。

第 5 行,stmt 是私有变量,用于执行 SQL 命令。

第 7 行到第 16 行,声明 connection()为公共方法,完成对数据库的连接。

第 9 行,装载 JDBC 驱动程序。

第 10 行,conn 被赋值为连接数据库 denglu 的语句。

第 11 行,stmt 被赋值为能够执行对 denglu 数据库操作的对象。

第 17 行到第 24 行,声明 query()方法是公共方法,完成数据库的查询功能,query()方法的返回值类型是 ResultSet,也就是说返回值为数据库里面的记录集。

第 25 行到第 32 行,声明 update()方法是公共方法,完成的功能是对数据库记录的更新,返回值是 int 型,如果返回值为 1 则说明更新数据库成功,如果不为 1 就说明更新失败。

第 33 行到第 40 行,声明 delete()方法是公共方法,完成的功能是对数据库记录的删除,返回值是布尔型。

第 41 行到第 49 行,声明 hzsb()方法是公共方法,完成的功能是汉字识别,返回值是字符串类型。

【源代码】

【denglu.html】

```
<html>
  <head>
    <title>登录页面</title>
    <style type="text/css">
      <!--
        .STYLE9 {font-family: Arial, Helvetica, sans-serif; font-size: 18px; }
      -->
    </style>
  </head>
  <body>
    <form id="form5" name="form5" method="post" action="denglu.jsp">
      <table width="500" border="0" align="center" cellpadding="0" cellspacing="2" bgcolor="#0099CC">
        <tr>
          <td width="203" align="right" valign="middle" bgcolor="#FFFFFF">
```

```
                <span class="STYLE9">用户名：</span></td>
15              <td width="394" height="50" valign="middle" bgcolor="#FFFFFF">
                     
16                <input name="username" type="text" id="textfield" size="15" /></td>
17            </tr>
18            <tr>
19              <td align="right" valign="middle" bgcolor="#FFFFFF"><span class=
                "STYLE9">密码：</span></td>
20              <td height="50" valign="middle" bgcolor="#FFFFFF">  
                   
21                <input name="password" type="password" id="textfield2" size="15" />
                </td>
22            </tr>
23            <tr>
24              <td height="50" colspan="2" align="center" valign="middle"bgcolor="#FFFFFF">
25                <input type="submit" name="button" id="button" value="登录" />
26                   
27                <input type="reset" name="button2" id="button2" value="取消" /></td>
28            </tr>
29            <tr>
30              <td height="50" colspan="2" align="center" valign="middle"
              bgcolor="#FFFFFF" class="STYLE9"><a href="zc.html">注册</a></td>
31            </tr>
32          </table>
33        </form>
34      </body>
35    </html>
```

denglu.html 补充说明

第 4 行到第 8 行，设置整个界面的字体格式。

第 11 行到第 33 行，是一个 form 表单，表单的数据将提交到 denglu.jsp 文件。

第 12 行到第 32 行，是一个表格；第 13 到第 17 行，是表格第一行，其中第二个单元格中放置了一个文本框，用于输入用户名，变量名定为 username；第 18 到第 22 行，是表格第二行，其中第二个单元格中放置了一个密码框，用于输入登录密码，变量名定为 password；第 23 行到第 28 行，是表格的第三行，其中放置了按钮，一个用于提交表单信息，一个用于重新输入；第 29 行到第 31 行，是表格第四行，单元格中设计了一个用户注册的链接，链接到 zc.html 文件。

【源代码】

【denglu.jsp】

```
<%@ page contentType="text/html; charset=gb2312" language="java" import="java.sql.*"%>
<html>
  <body>
    <jsp:useBean id="db" class="bean.mydb" scope="page"/>
    <%
      String username=db.hzsb(request.getParameter("username"));
```

```
7          String password=request.getParameter("password");
8          db.connection();
9          ResultSet rs=null;
10         rs=db.query("SELECT * FROM customer WHERE id="+"\'"+username+"\'"+ "
           AND password="+"\'"+password+"\'");
11         while(rs.next())
12         {
13             session.setAttribute("username",username);
14             response.sendRedirect("success.jsp");
15         }
16       %>
17     </body>
18   </html>
```

denglu.jsp 补充说明

第 1 行，引入 Java 类的 SQL 包，因为下面代码用到了 SQL 命令。

第 4 行，使用 useBean 动作，使用的类是 bean 文件夹下的 mydb 类，类的对象为 db。

第 6 行，新建字符串型变量 username 接收 denglu.html 文件传递过来的用户名，并且使用 db 对象调用 hzsb()方法，以防用户输入汉字产生乱码。

第 7 行，新建字符串型变量 password 接收 denglu.html 文件传递过来的密码。

第 8 行，使用 db 对象调用 connection()方法，连接数据库。

第 9 行，声明 rs 为记录集型的变量。

第 10 行，db 对象调用 query()方法查找数据库的 customer 表，看前面接收的用户名和密码是否与表中的符合，并把 query()方法的返回值赋给 rs。

第 11 行到第 15 行，当 rs 中有符合上面查找条件的记录集时，使用 session 方法把用户登录时输入的用户名赋值给 username 变量，然后使用 response 方法直接把链接地址定向到登录成功页面 success.jsp。

【源代码】

【zc.html】

```
<html>
  <head>
    <title>会员注册</title>
    <style type="text/css">
      <!--
        .STYLE1 { font-family: "华文新魏"; font-size: 24px; color: #FF9900;}
        .STYLE4 {font-family: "宋体"; font-size: 18px; }
      -->
    </style>
  </head>
  <body>
    <form id="form1" name="form1" method="post" action="zc_1.jsp">
      <table width="400" border="0" align="center" cellpadding="0" cellspacing="2"
      bgcolor="#CCCCCC">
        <tr>
          <td height="40" colspan="2" align="center" valign="middle" bgcolor=
```

```
      "#FFFFFF"><span class="STYLE1">用户注册</span></td>
    </tr>
    <tr>
      <td width="40%" height="30" align="center" bgcolor="#FFFFFF"><span class="STYLE4">用户名：</span></td>
      <td height="30" align="left" bgcolor="#FFFFFF">
        <input type="text" name="username" id="textfield" /></td>
    </tr>
    <tr>
      <td height="30" align="center" bgcolor="#FFFFFF"><span class="STYLE4">密码：</span></td>
      <td height="30" align="left" bgcolor="#FFFFFF">
        <input type="text" name="password" id="textfield2" /></td>
    </tr>
    <tr>
      <td height="30" align="center" bgcolor="#FFFFFF"><span class="STYLE4">确认密码：</span></td>
      <td height="30" align="left" bgcolor="#FFFFFF">
        <input type="text" name="confirmpassword" id="textfield3" /></td>
    </tr>
    <tr>
      <td height="30" align="center" bgcolor="#FFFFFF"><span class="STYLE4">姓名：</span></td>
      <td height="30" align="left" bgcolor="#FFFFFF">
        <input type="text" name="name" id="textfield4" /></td>
    </tr>
    <tr>
      <td height="30" align="center" bgcolor="#FFFFFF"><span class="STYLE4">性别：</span></td>
      <td height="30" align="left" bgcolor="#FFFFFF">
        <select name="sex" id="select">
          <option>男</option>
          <option>女</option>
        </select></td>
    </tr>
    <tr>
      <td height="30" align="center" bgcolor="#FFFFFF"><span class="STYLE4">地址：</span></td>
      <td height="30" align="left" bgcolor="#FFFFFF">
        <input type="text" name="address" id="textfield6" /></td>
    </tr>
    <tr>
      <td height="30" align="center" bgcolor="#FFFFFF"><span class="STYLE4">邮编：</span></td>
      <td height="30" align="left" bgcolor="#FFFFFF">
        <input type="text" name="code" id="textfield7" /></td>
    </tr>
    <tr>
      <td height="30" align="center" bgcolor="#FFFFFF"><span class="STYLE4">
```

```
                "STYLE4">电话：</span></td>
57              <td height="30" align="left" bgcolor="#FFFFFF">
58                <input type="text" name="tel" id="textfield8" /></td>
59            </tr>
60            <tr>
61              <td height="30" align="center" bgcolor="#FFFFFF"><span class=
                "STYLE4">e-mail：</span></td>
62              <td height="30" align="left" bgcolor="#FFFFFF">
63                <input type="text" name="email" id="textfield9" /></td>
64            </tr>
65            <tr>
66              <td height="30" colspan="2" align="center" valign="middle" bgcolor=
                "#FFFFFF">
67                <input type="submit" name="button" id="button" value="确定" />
                      
68                <input type="reset" name="button2" id="button2" value="重填" /></td>
69            </tr>
70          </table>
71        </form>
72      </body>
73  </html>
```

zc.html 补充说明

第 4 行到第 9 行，设置界面的字体格式设置。

第 12 到第 71 行，是一个表单，表单的数据将提交到 zc_1.jsp 文件。

第 13 到第 70 行，是一个表格；第 14 行到第 16 行是表格的第一行；第 17 行到第 21 行，是表格的第二行，在这行的第二个格里有一个文本框，用于输入注册的用户名，变量名为 username；第 22 行到第 26 行，是表格的第三行，在这行的第二个格里有一个文本框，用于输入注册密码，变量名为 password；第 27 行到第 31 行，是表格的第四行，在这行的第二个格里有一个文本框，用于再次输入注册的密码，变量名为 confirmpassword；第 32 行到第 36 行，是表格的第五行，在这行的第二个格里有一个文本框，用于输入用户真实姓名，变量名为 name；第 37 行到第 44 行，是表格的第六行，在这行的第二个格里有一个下拉列表，下拉列表有两个选项，变量名为 sex，用于选择注册用户的性别；第 45 行到第 49 行，是表格的第七行，在这行的第二个格里有一个文本框，用于输入用户住址，变量名为 address；第 50 行到第 54 行表格的第八行，在这行的第二个格里有一个文本框，用于输入用户邮编，变量名为 code；第 55 行到第 59 行表格的第九行，在这行的第二个格里有一个文本框，用于输入用户电话，变量名为 tel；第 60 行到第 64 行表格的第十行，在这行的第二个格里有一个文本框，用于输入用户邮件，变量名为 email；第 65 行到第 69 行表格的第十一行，在这行里放置两个按钮，一个用于提交表单的所有数据，一个用于重新填写表单内容。

【源代码】

【zc_1.jsp】

```
<%@ page contentType="text/html; charset=gb2312" language="java" import="java.sql.*"%>
<html>
```

```
3     <body>
4       <jsp:useBean id="db" class="bean.mydb" scope="page"/>
5       <%
6         String id=db.hzsb(request.getParameter("username")) ;
7         String password=db.hzsb(request.getParameter("password"));
8         String name=db.hzsb(request.getParameter("name"));
9         String sex=db.hzsb(request.getParameter("sex"));
10        String address=db.hzsb(request.getParameter("address"));
11        String code=db.hzsb(request.getParameter("code"));
12        String tel=db.hzsb(request.getParameter("tel"));
13        String email=db.hzsb(request.getParameter("email"));
14        if (id!=null && password!=null && name!=null && sex!=null &&
          address!=null&& code!=null && tel!=null && email!=null)
15        {   String rssql="insert into customer values("+"\'"+id+"\'"+","+"\'"+
              password+"\'"+","+"\'"+name+"\'"+","+"\'"+sex+"\'"+","+"\'"+
              address+"\'"+","+"\'"+code+"\'"+","+"\'"+tel+"\'"+","+"\'"+email
              +"\'"+")";
16            db.connection();
17            int rsn=db.update(rssql);
18            if (rsn>0)
19              response.sendRedirect("denglu.html");
20        }%>
21    </body>
22  </html>
```

zc_1.jsp 补充说明

第 1 行，引入 java 类的 SQL 包，因为下面代码用到了 SQL 命令。

第 4 行，使用 useBean 动作，使用的类是 bean 文件夹下的 mydb 类，类的对象为 db。

第 6 行到第 13 行，接收 zc.html 文件传递过来的各项注册信息，并且使用 db 对象调用 hzsb()方法，以防用户输入汉字产生乱码。

第 14 行到第 20 行，是一组 if 语句，当所有注册信息都不为空的时候执行 if 语句中的所有操作。

第 15 行，声明一个字符串型变量 rssql，赋值为 SQL 插入语法的句子。

第 16 行，使用 db 对象调用 connection()方法连接数据库。

第 17 行，使用 db 对象调用 update()方法执行对 customer 表的记录插入，并把返回值赋给整形变量 rsn。

第 18 行到第 19 行，是一组 if 语句，如果 rsn 的返回值大于 0，就表示上一句的记录插入成功，那么就可以直接跳转到 denglu.html 界面，进行登录了。

【源代码】

【success.jsp】

```
<%@ page contentType="text/html; charset=gb2312" language="java" %>
<html>
  <head>
```

```
        <title>登录成功</title>
    </head>
    <body>
        欢迎   <%=session.getAttribute("username")%>  登录
    </body>
</html>
```

5.2.1 JDBC技术和驱动程序

JDBC(Java DataBase Connectivity)可以说是 Java 应用程序与数据库的沟通桥梁。通过 JDBC 提供的 API(应用程序接口)进行程序设计时，可以轻松地编写出和数据库连接的应用程序。

5.2.2 JSP中使用JDBC—ODBC访问数据库

JDBC-ODBC 驱动程序是 JDBC 连接数据的方法之一，该驱动程序首先将 JDBC 数据转换成 ODBC 数据源，然后再利用 ODBC 与数据库进行连接。

在 JSP 中使用 JDBC—ODBC 访问数据库一般来说有下列几个步骤。

1. 装载数据库的 JDBC 驱动程序

装载数据库系统的驱动程序，是将 JDBC 驱动程序的类载入到 Java 虚拟机中，需要 java.lang.class 类中的 Class.forName()方法来实现。

语法格式：

```
Class.forName( driver_name );
```

上面例题中的用法：

```
Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
```

2. 与数据库建立连接

Connection 是用来表示数据库连接的对象，对数据库的一切操作都是在这个连接的基础上进行的。

语法格式：

```
Connection con=DriverManager.getConnection(url, username,password)
```

上面例题中的用法：

```
conn=DriverManager.getConnection("jdbc:odbc:shici");
```

3. 将 SQL 语句传递到数据库中，执行操作，返回得到的结果

Statement 对象是一个接口的定义，其中包括了执行 SQL 语句和获取返回结果的方法。Statement 对象用于执行不带参数的简单 SQL 语句。

语法格式：

```
Statement stmt = con.createStatement();
```

上面例题中的用法：

```
stmt=conn.createStatement();
```

Statement 接口提供了 3 种执行 SQL 语句的方法。

(1) executeQuery()方法用于产生单个结果集的 SQL 查询语句

上面例题中的用法：

```
ResultSet rs=stmt.executeQuery(sql);
```

ResultSet 用来暂时存放数据库查询操作获得的结果。它包含了符合 SQL 语句中条件的所有行，并提供了 get 方法对这些行中的数据进行访问。

(2) executeUpdate()方法返回值是一个整型值

上面例题中的用法：

```
n=stmt.executeUpdate(sql);
```

(3) execute()方法用于执行返回多个结果集或多个更新计数的语句，返回值是布尔值

上面例题中的用法：

```
b=stmt.execute(sql);
```

4. 关闭数据库连接

语法格式 rs.close();

5.3 JSP 对数据库的操作

实战演练 1——诗词目录

【题目】

要求制作一个诗词目录，能够显示诗词的序号、作者、类别、题目，并且能够增加、修改、删除任意一条目录，只有诗词目录的序号不可更改。

【分析】

要完成本题目需要先理清思路一步一步地完成，步骤如下。

(1) 本题要完成的功能模块如图 5-44 所示。

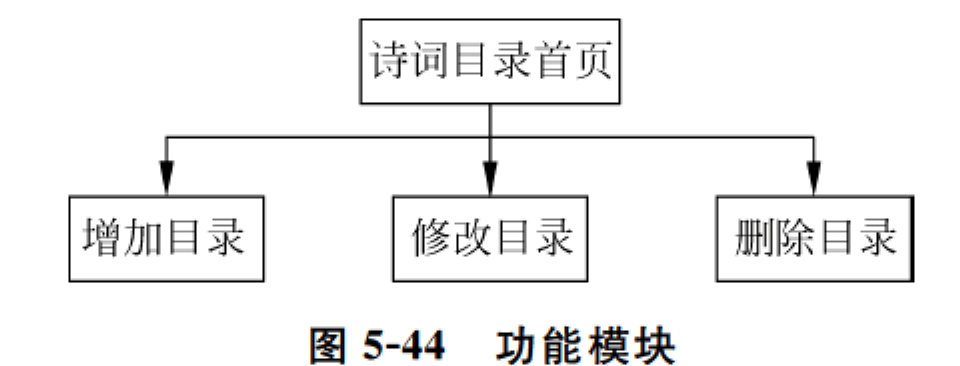


图 5-44 功能模块

（2）打开 SQL 软件的企业管理器，新建数据库 shici，如图 5-45 所示。

图 5-45　数据库 shici 表 mulu

（3）在 shici 数据库中新建会员用户表 mulu，表的设计情况如图 5-46 所示。



列名	数据类型	长度	允许空	备注
id	varchar	15		序号，设置为主键
name	varchar	10		诗词作者
lei	varchar	10		诗词类别
timu	varchar	30		诗词题目

图 5-46　mulu 表的设计和说明

（4）建立 ODBC 数据源，打开系统管理工具——ODBC 数据源对话框，添加 shici 数据源，如图 5-47 所示。

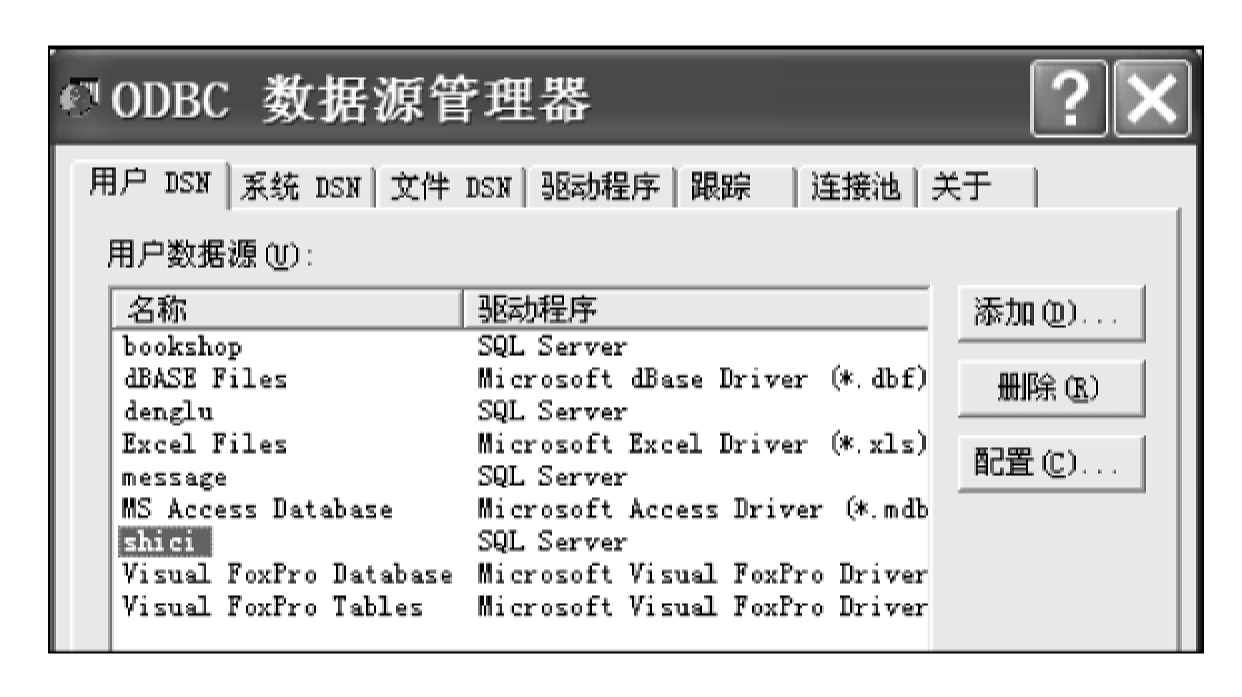


图 5-47　添加数据源 shici

（5）题目中需要的代码文件有：mydb.java 完成保存连接数据库的代码和用于对数据库进行增删改查以及汉字识别的方法；shici.jsp 完成对数据库中 mulu 表的读取，并显示在界面上，还要提供“添加”、“删除”、“修改”的链接；sc_add.html 完成添加目录的界面设计，并且提交输入的数据；sc_add_1.jsp 完成对目录信息的入库操作；sc_edit.jsp 完成需要修改的目录信息的显示，并且把修改后的目录信息更新到 mulu 表中；sc_del.jsp 完成对某条目录的删除。

【运行效果】

打开开发工具 MyEclipse，在工程 jsp-jdbc 中，在 bean 文件夹中修改 mydb.java 文

件；在这个工程的 WebRoot 文件夹下，新建文件 shici. jsp、sc_add. html、sc_add_1. jsp、sc_edit. jsp、sc_del. jsp。程序输入完毕并且保存之后，运行开发工具 MyEclipse 中的 Tomcat 服务器。最后在浏览器中输入地址 http://127. 0. 0. 1:8080/jsp-jdbc/shici. jsp，就可以看到运行结果了(如图 5-48～图 5-52 所示)。



图 5-48　首页面 shici. jsp



图 5-49　添加页面 sc_add. html　　**图 5-50　修改页面 sc_edit. jsp**



图 5-51　添加成功返回首页面

【源代码】

【mydb. java】

```
package bean;
import java.sql. * ;
```

第一个数据库练习

唐诗宋词目录

序号	作者	类别	题目		
1	李白	唐诗	将进酒	修改	删除
2	王维	唐诗	鸟鸣涧	修改	删除
3	杜牧	唐诗	夜雨寄北	修改	删除
4	李白	唐诗	月下独酌	修改	删除
5	柳宗元	唐诗	渔翁	修改	删除

添加

图 5-52 修改成功返回首页面 sc_add.jsp

```
3   public class mydb {
4   …
5   public void connection() {
6       try{
7           Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
8           conn=DriverManager.getConnection("jdbc:odbc:shici");
9   …
```

mydb.java 补充说明：

这个文件和上一个题目的文件是一样的，只需将第 8 行代码中数据库的名字改成 shici 就可以通用这整篇代码了。

【源代码】

【shici.jsp】

```
<%@page contentType="text/html; charset=gb2312" import="java.sql.*" %>
<html><head><title>第一个数据库练习<title><head>
  <body>
    <jsp:useBean id="db" class="bean.mydb" scope="page"/>
    <p align="center">唐诗宋词目录</p>
    <table width="700" cellspacing="1" cellpadding="8" border="0"bgcolor="#669966"
    align="center" height="75">
      <tr>
         <td bgcolor="#FFFFFF" width="10%"> 序号</td>
         <td bgcolor="#FFFFFF" width="20%">作者  </td>
         <td bgcolor="#FFFFFF" width="10%">类别  </td>
         <td bgcolor="#FFFFFF" width="40%">题目  </td>
         <td colspan="2" bgcolor="#FFFFFF" width="20%"> </td>
      </tr>
      <%
        ResultSet rs=null;
        db.connection();
        String query="select * from mulu";
        rs=db.query(query);
        while (rs.next())
        {
        String s1=rs.getString(1);
```

```
22          String s2=rs.getString(2);
23          String s3=rs.getString(3);
24          String s4=rs.getString(4);
25        %>
26        <tr>
27          <td bgcolor="#FFFFFF"><%=s1%><br></td>
28          <td bgcolor="#FFFFFF"><%=s2%><br></td>
29          <td bgcolor="#FFFFFF"><%=s3%><br></td>
30          <td bgcolor="#FFFFFF"><%=s4%><br></td>
31          <td bgcolor="#FFFFFF"> <a href="sc_edit.jsp?id=<%=s1 %> ">
            修改</a></td>
32          <td bgcolor="#FFFFFF"> <a href="sc_del.jsp?id=<%=s1 %> ">
            删除</a></td>
33        </tr>
34        <%  }     %>
35      </table>
36      <p align="center"><a href="sc_add.jsp">添加
37    </body>
38  </html>
```

shici.jsp 补充说明

第 1 行，引入 java 类的 SQL 包，因为下面代码用到了 SQL 命令。

第 4 行，使用 useBean 动作，使用的类是 bean 文件夹下的 mydb 类，类的对象为 db。

第 6 行到第 35 行，是一个表格，第 7 行到第 13 行，是表格的第一行。

第 15 行，声明 rs 为记录集型的变量。

第 16 行，使用 db 对象调用 connection()方法，连接数据库。

第 17 行，声明一个字符串型变量 query，赋值为 mulu 表查找语句。

第 18 行，db 对象调用 query()方法执行查找 mulu 表的语句，然后把记录集的返回值赋值给 rs。

第 19 行到第 34 行，是一个 while 循环，循环条件是 rs.next()，直到把 rs 中的所有记录信息都读出来才结束循环；第 21 行到第 24 行，把记录集里面的目录信息使用 get 方法取出来赋值给字符串型变量。

第 26 行到第 33 行，是表格的第二行，把 rs 记录集里面取出的目录信息放置到这行的前四个单元格中，第五个单元格和第六个单元格放置目录信息相对应的“修改”和“删除”链接。

第 36 行，单独一个段落放置“添加”链接。

【源代码】

【sc_edit.jsp】

```
<%@page contentType="text/html; charset=gb2312" import="java.sql.*" %>
<html>
  <head><title>向数据库修改目录信息</title></head>
    <jsp:useBean id="db" class="bean.mydb" scope="page"/>
  <body>
    <%
```

```
        String id=request.getParameter("id");
        ResultSet rs=null;
        if (id!=null)
        {   String sql="SELECT * FROM mulu WHERE id='"+id+"'";
            db.connection();
            rs=db.query(sql);
            while (rs.next())
        {   String s1=rs.getString(1);
            String s2=rs.getString(2);
            String s3=rs.getString(3);
            String s4=rs.getString(4);
%>
<div align="center">
<form name="form1" id="form1" method="post" action="">
  <p>修改目录 </p>
    <table width="323" border="0" cellpadding="8" cellspacing="1" bgcolor="#669966">
      <tr>
        <td width="30%" bgcolor="#FFFFFF" align="center">序号: </td>
        <td width="70%" bgcolor="#FFFFFF">
          <input name="id" type="text" id="id" value="<%=s1%>" disabled></td>
      </tr>
      <tr>
        <td bgcolor="#FFFFFF" align="center">作者: </td>
        <td bgcolor="#FFFFFF">
          <input name="name" type="text" id="name" value="<%=s2%>"></td>
      </tr>
      <tr>
        <td bgcolor="#FFFFFF" align="center">类别:</td>
        <td bgcolor="#FFFFFF"><input name="lei" type="text" id="lei" value="<%=s3%>"></td>
      </tr>
      <tr>
        <td bgcolor="#FFFFFF" align="center">题目:</td>
        <td bgcolor="#FFFFFF"><input name="timu" type="text" id="timu" value="<%=s4%>"></td>
      </tr>
      <tr>
        <td colspan="2" align="center" valign="middle" bgcolor="#FFFFFF">
          <input type="submit" name="Submit" value="确定" ></td>
      </tr>
    </table>
  </form></div>
<%}}
  id=request.getParameter("id");
  String name=db.hzsb(request.getParameter("name"));
  String lei=db.hzsb(request.getParameter("lei"));
```

```
51          String timu=db.hzsb(request.getParameter("timu"));
52          if (id!=null && name!=null && lei!=null && timu!=null)
53            { String sql="update mulu set name='"+name+"',lei='"+lei+"',timu='"+
                timu+"' where id='"+id+"'";
54              db.connection();
55              int rsn=db.update(sql);
56              if (rsn>0)
57              response.sendRedirect("shici.jsp");
58      }%>
59    </body>
60  </html>
```

sc_edit.jsp 补充说明

第 7 行，接收 shici.jsp 页面传递过来的 id 信息，也就是目录所在 mulu 表的序号。

第 9 行，if 语句，如果 id 不为空就执行下面的语句。

第 10 行，是一个 sql 查找语句，是根据 id 的值查找 mulu 表中复合的记录集。

第 13 到第 47 行，是 while 循环，循环条件是 rs.next()直到把 rs 中的所有记录信息都读出来才结束循环；第 14 行到第 17 行，把记录集里面的目录信息使用 get 方法取出来赋值给字符串型变量 s1、s2、s3、s4；第 20 行到第 46 行，是一个 form 表单；第 22 行到第 45 行，是一个表格，第 23 行到第 27 行是表格的第一行，在这行的第二个单元格中放置一个文本框，文本框的变量名为 id，默认值为 s1，而且文本框被设定为不可修改；第 28 行到第 32 行是表格的第二行，在这行的第二个单元格中放置一个文本框，文本框的变量名为 name，默认值为 s2；第 33 行到第 36 行是表格的第三行，在这行的第二个单元格中放置一个文本框，文本框的变量名为 lei，默认值为 s3；第 37 行到第 40 行是表格的第四行，在这行的第二个单元格中放置一个文本框，文本框的变量名为 timu，默认值为 s4；第 41 行到第 44 行是表格的第五行，在这行放置一个提交按钮，用于提交修改后的目录信息。

第 48 到第 51 行，接收前面表单提交的已修改的各项目录信息。

第 52 行，if 语句判断如果表单提交的各项信息都不为空，那么就执行下面(第 53 到第 57 行)的语句。

第 53 行，更新数据库 mulu 表中各项信息的 sql 语句。

第 54 行，db 对象调用 connection()方法链接数据库；第 55 行，db 对象调用 update()方法执行数据库的更新。

第 56 行到第 57 行，根据更新数据库后的返回值 rsn 判断更新是否成功，如果更新成功就完成页面的跳转，跳转到 shici.jsp 页面。

【源代码】

【sc_add.html】

```
<html>
  <head><title>向数据库添加目录信息</title></head>
  <body>
    <div align="center">
      <form name="form1" id="form1" method="post" action="sc_add_1.jsp">
        <p>添加目录</p>
```

```
7            <p>要求新添加的目录内容序号不能和已存在的重复</p>
8            <table width="323" border="0" cellpadding="8" cellspacing="1" bgcolor=
             "#669966">
9              <tr>
10               <td width="30%" bgcolor="#FFFFFF" align="center">序号：</td>
11               <td width="70%" bgcolor="#FFFFFF">
12                 <input name="id" type="text" id="id" ></td>
13             </tr>
14             <tr>
15               <td bgcolor="#FFFFFF" align="center">作者：</td>
16               <td bgcolor="#FFFFFF"><input name="name" type="text" id="name" >
                 </td>
17             </tr>
18             <tr>
19               <td bgcolor="#FFFFFF" align="center">类别:</td>
20               <td bgcolor="#FFFFFF"><input name="lei" type="text" id="lei" ></td>
21             </tr>
22             <tr>
23               <td bgcolor="#FFFFFF" align="center">题目:</td>
24               <td bgcolor="#FFFFFF"><input name="timu" type="text" id="timu" >
                 </td>
25             </tr>
26             <tr>
27               <td colspan="2" align="center" valign="middle" bgcolor="#FFFFFF">
28                 <input type="submit" name="Submit" value="确定" ></td>
29             </tr>
30           </table>
31         </form></div>
32     </body>
33 </html>
```

sc_add.html 补充说明

第 5 行到第 31 行，是一个表单，表单将把数据提交到 sc_add_1.jsp 文件；第 8 行到第 30 行，是一个表格，第 9 行到第 13 行，是表格的第一行，在这行的第二个单元格放置一个文本框，变量名为 id，用于输入目录的序号；第 14 行到第 17 行，是表格的第二行，在这行的第二个单元格放置一个文本框，变量名为 name，用于输入诗词的作者；第 18 行到第 21 行，是表格的第三行，在这行的第二个单元格放置一个文本框，变量名为 lei，用于输入诗词的类别；第 22 行到第 25 行，是表格的第四行，在这行的第二个单元格放置一个文本框，变量名为 timu，用于输入诗词的题目；第 26 行到第 29 行，是表格的第五行，在这行里放置了一个提交按钮。

【源代码】

【sc_add_1.jsp】

```
<%@ page contentType="text/html;charset=gb2312" import="java.sql.*"%>
<html>
  <body>
  <jsp:useBean id="db" class="bean.mydb" scope="page"/>
    <%String id=request.getParameter("id");
      String name=db.hzsb(request.getParameter("name"));
```

```
7           String lei=db.hzsb(request.getParameter("lei"));
8           String timu=db.hzsb(request.getParameter("timu"));
9           if (id!=null && name!=null && lei!=null && timu!=null)
10          {String sql="INSERT INTO mulu VALUES("+"\'"+id+"\'"+","+"\'"+name+
            "\'"+","+"\'"+lei+"\'"+","+"\'"+timu+"\'"+")";
11          db.connection();
12          int rsn=db.update(sql);
13          if (rsn>0)
14              response.sendRedirect("shici.jsp");
15          }%>
16      </body>
17  </html>
```

sc_add_1.jsp 补充说明

第 1 行，引入 java 类的 SQL 包，因为下面代码用到了 SQL 命令。

第 4 行，使用 useBean 动作，使用的类是 bean 文件夹下的 mydb 类，类的对象为 db。

第 5 行到第 8 行，接收 sc_add.html 文件传递过来的各项诗词信息，并且使用 db 对象调用 hzsb()方法，以防用户输入汉字产生乱码。

第 9 行到第 15 行，是一组 if 语句，当所有诗词目录信息都不为空的时候执行 if 语句中的所有操作。

第 10 行，声明一个字符串型变量 SQL，赋值为向 mulu 表插入诗词信息的句子。

第 11 行，使用 db 对象调用 connection()方法连接数据库。

第 12 行，使用 db 对象调用 update()方法执行对 mulu 表的记录插入，并把返回值赋给整形变量 rsn。

第 13 行到第 14 行，是一组 if 语句，如果 rsn 的返回值大于 0，就表示上一句的记录插入成功，那么就可以直接跳转到 shici.jsp 界面。

【源代码】

【sc_del.jsp】

```
<%@page contentType="text/html; charset=gb2312" import="java.sql.*" %>
<html>
  <head>
    <title>删除指定目录信息</title>
  </head>
  <jsp:useBean id="db" class="bean.mydb" scope="page"/>
  <body>
    <%
      String id=request.getParameter("id");
      if (id!=null)
      {
          String sql="DELETE FROM mulu WHERE id="+"\'"+id+"\'";
          db.connection();
          db.delete(sql);
          response.sendRedirect("shici.jsp");
      }
```

```
    %>
  </body>
</html>
```

实战演练 2——图书库存情况的查找

【题目】

要求在界面上显示全部图书的库存信息情况，并且可以根据图书的序号、书名、作者、出版社、类型、单价、库存查找书籍。

【分析】

(1) 打开 SQL 软件的企业管理器，新建数据库 chazhao，如图 5-53 所示。

(2) 在 chazhao 数据库中新建库存表 book，表的设计情况如图 5-54 所示。



图 5-53　新建数据库 chazhao

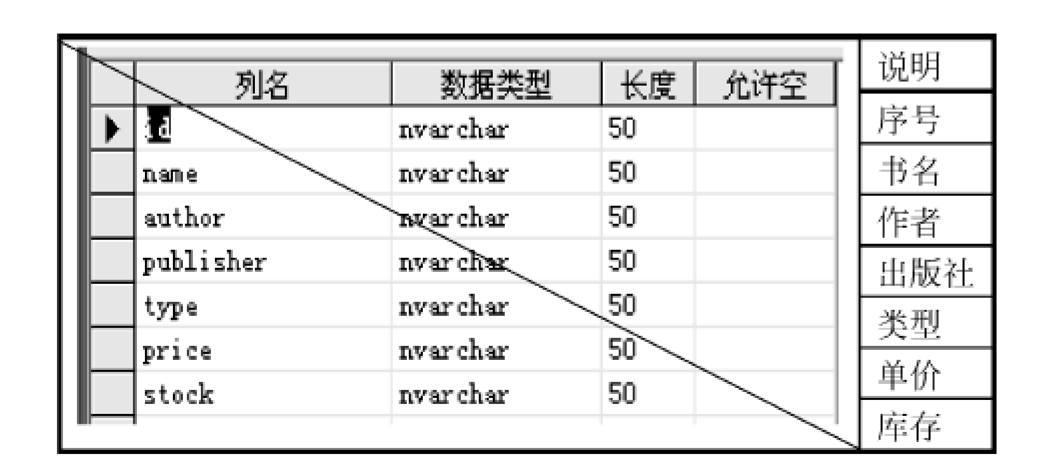


图 5-54　book 表的设计

(3) 建立 ODBC 数据源，打开系统管理工具——ODBC 数据源对话框，添加 chazhao 数据源，如图 5-55 所示。

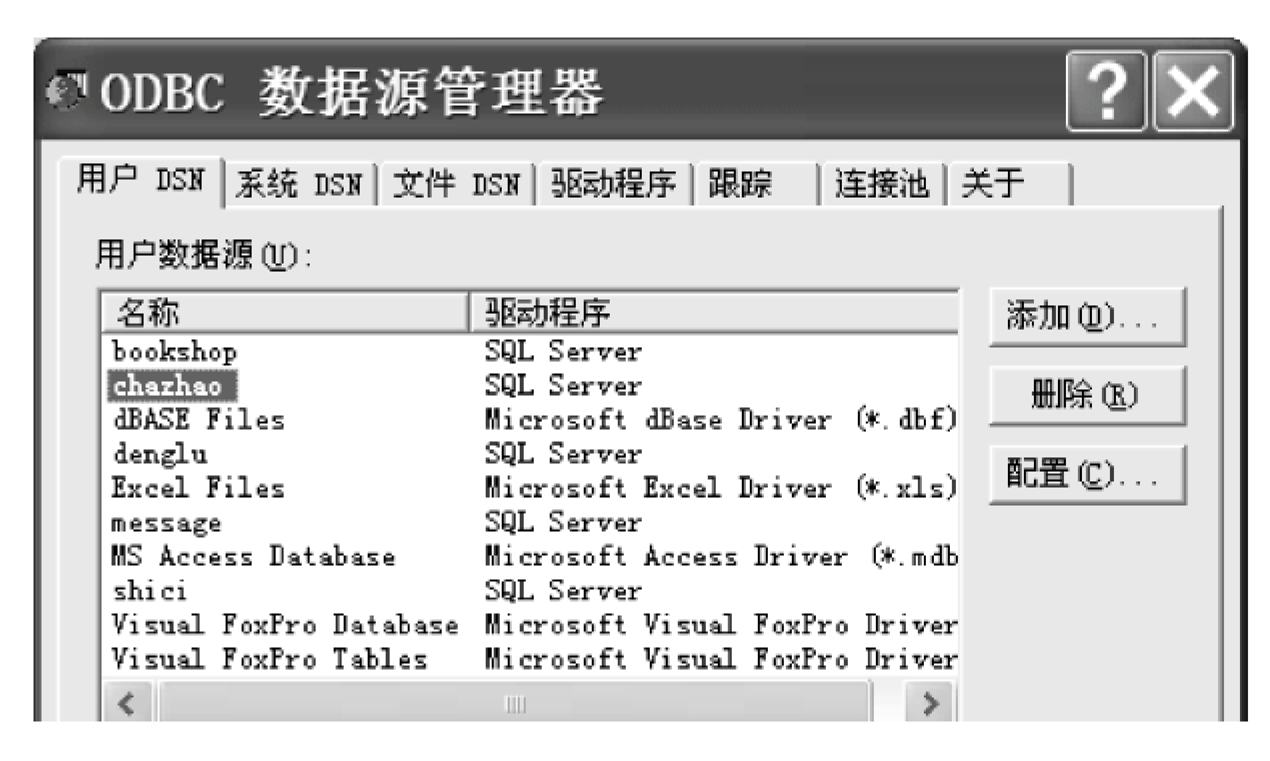


图 5-55　添加 chazhao 数据源

(4) 题目中需要的代码文件有：mydb.java 完成保存连接数据库的代码和用于对数据库进行增删改查以及汉字识别的方法；chaz.jsp 文件完成查找项目的表格设计，并且

把 chaz_1.jsp 文件的内容引入进来，使两个文件的显示内容合二为一；chaz_1.jsp 文件可以显示图书库存的所有情况；result.jsp 文件设计了图书的查找结果的表格，并且把 result_1.jsp 文件的内容引入进来，使查找结果显示在表格中；result_1.jsp 文件显示图书的查找结果。

【运行效果】

打开开发工具 MyEclipse，在工程 jsp-jdbc 的 bean 文件夹中修改 mydb.java 文件。在这个工程的 WebRoot 文件夹下，新建文件 chaz.jsp、chaz_1.jsp、result.jsp、result_1.jsp。程序输入完毕并且保存之后，运行开发工具 MyEclipse 中的 Tomcat 服务器。最后在浏览器中输入地址 http://127.0.0.1:8080/jsp-jdbc/chaz.jsp，就可以看到运行结果了（如图 5-56 和图 5-57 所示）。

图书管理》》查询图书

图书ID：		图书名称：		作者：	
出版社：		类别：	科技		查找

全部图书如下

图书ID	图书名称	作者	出版社	类别	价格	库存
1	计算机基础	xx	高教出版社	教育	20	4
2	财富	gg	上海	生活	20	7
3	旅行者	ss	上海	生活	15	10
4	绝对小孩儿	nn	文艺出版社	娱乐	23	8
5	网络技术	ww	人民邮电出版社	教育	12	10
6	瑜伽	tt	人民邮电出版社	生活	34	6
7	广告创意	ww	机械工业出版社	科技	44	45
8	JSP	JSP	JSP	科技	30	66

图 5-56　chaz.jsp

图书ID	图书名称	作者	出版社	类别	价格	库存
1	计算机基础	xx	高教出版社	教育	20	4
5	网络技术	ww	人民邮电出版社	教育	12	10

返回所有图书首页

图 5-57　result.jsp

【源代码】

【mydb.java】

```
package bean;
import java.sql.*;
public class mydb {
…
public void connection() {
      try{
          Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
          conn=DriverManager.getConnection("jdbc:odbc:chazhao");
…
```

mydb. java 补充说明

这个文件和上一个题目的文件是一样的，只需将第 8 行代码中数据库的名字改成 chazhao 就可以通用这整篇代码了。

【源代码】

【chaz. jsp】

```
<%@page contentType="text/html; charset=gb2312" import="java.sql.*"%>
<html>
  <head><title>查找功能</title></head>
  <body>
    <form id="form1" name="form1" method="post" action="result.jsp">
      <table width="900" border="1" align="center" cellpadding="0" cellspacing="0">
        <tr>
            <td height="30" colspan="6" align="left" valign="middle" bgcolor="#CCCCCC">图书管理》》查询图书</td>
        </tr>
        <tr>
           <td height="15" align="right" valign="middle" width="80">图书 ID: </td>
           <td  align="left" valign="middle">
             <input name="bookid" type="text"  id="textfield" /></td>
           <td  align="right" valign="middle" width="80">图书名称: </td>
           <td  align="left" valign="middle">
             <input type="text" name="bookname" id="textfield2" /></td>
           <td  align="right" valign="middle" width="80">作者: </td>
           <td  align="left" valign="middle">
             <input type="text" name="author" id="textfield3" /></td>
        </tr>
        <tr>
           <td height="15" align="right" valign="middle" >出版社: </td>
           <td  align="left" valign="middle">
             <input type="text" name="publisher" id="textfield4" /> </td>
           <td  align="right" valign="middle" >类别: </td>
           <td  align="left" valign="middle">
             <select name="category" id="select">
               <option >null</option>
               <option >科技</option>
               <option>艺术</option>
               <option>生活</option>
               <option>社会</option>
               <option>娱乐</option>
               <option>教育</option></select></td>
           <td  colspan="2" align="center" valign="middle">
               <input type="submit" name="button" id="button" value="查找" /></td>
        </tr></table></form>
    <%@include file="chaz_1.jsp"%>
  </body>
</html>
```

chaz.jsp 补充说明

第 5 行到第 37 行，是一个表单，表单的数据将提交到 result.jsp；第 6 行到第 37 行，是一个表格；第 7 行到第 9 行，是表格的第一行；第 10 行到第 20 行，是表格的第二行，在这行的第二个格放置了一个文本框，变量名为 bookid，用于输入图书 ID，第四个格放置了一个文本框，变量名为 bookname，用于输入图书名称，第六个格放置了一个文本框，变量名为 author，用于输入书籍作者；第 21 到第 37 行，是表格的第三行，在这行的第二个格放置了一个文本框，变量名为 publisher，用于输入出版社，第四个格，放置一个下拉框，变量名为 category，第五个格放置了一个提交按钮；第 38 行，引入了 chaz_1.jsp 文件，可以在页面中显示图书库存的所有情况。

【源代码】

【chaz_1.jsp】

```
<%@page contentType="text/html; charset=gb2312" import="java.sql.* "%>
<html>
  <jsp:useBean id="db" class="bean.mydb" scope="page"/>
  <body>
    <br><h2>全部图书如下</h2>
    <table width="900" border="0" cellspacing="0" cellpadding="0">
      <tr>
        <td height="30" align="center" valign="middle" bgcolor="#CCCCCC">图书ID </td>
        <td height="30" align="center" valign="middle" bgcolor="#CCCCCC">图书名称  </td>
        <td height="30" align="center" valign="middle" bgcolor="#CCCCCC">作者 </td>
        <td height="30" align="center" valign="middle" bgcolor="#CCCCCC">出版社 </td>
        <td height="30" align="center" valign="middle" bgcolor="#CCCCCC">类别 </td>
        <td height="30" align="center" valign="middle" bgcolor="#CCCCCC">价格 </td>
        <td height="30" align="center" valign="middle" bgcolor="#CCCCCC">库存 </td>
      </tr>
      <%
        ResultSet rs=null;
        db.connection();
        String query="select * from book";
        rs=db.query(query);
        while (rs.next())
        { String s1=rs.getString(1);
          String s2=rs.getString(2);
          String s3=rs.getString(3);
          String s4=rs.getString(4);
          String s5=rs.getString(5);
          String s6=rs.getString(6);
```

```
                String s7=rs.getString(7); %>
            <tr>
              <td height="30" align="center" valign="middle"><%=s1%></td>
              <td height="30" align="center" valign="middle"><%=s2%></td>
              <td height="30" align="center" valign="middle"><%=s3%></td>
              <td height="30" align="center" valign="middle"><%=s4%></td>
              <td height="30" align="center" valign="middle"><%=s5%></td>
              <td height="30" align="center" valign="middle"><%=s6%></td>
              <td height="30" align="center" valign="middle"><%=s7%></td>
            </tr>
            <%   } %></table>
    </body>
</html>
```

【result. jsp】

```
<%@ page contentType="text/html; charset=gb2312" import="java.sql.*"%>
<html >
  <head><title>查询结果</title></head>
  <jsp:useBean id="db" class="bean.mydb" scope="page"/>
  <body>
    <center>
      <form id="form1" name="form1" method="post" action="">
        <table width="700" border="0" cellspacing="0" cellpadding="0">
          <tr>
            <td height="30" align="center" valign="middle" bgcolor="#CCCCCC">
             图书 ID</td>
            <td height="30" align="center" valign="middle" bgcolor="#CCCCCC">图书
            名称  </td>
            <td height="30" align="center" valign="middle" bgcolor="#CCCCCC">作者
             </td>
            <td height="30" align="center" valign="middle" bgcolor="#CCCCCC">出版
            社  </td>
            <td height="30" align="center" valign="middle" bgcolor="#CCCCCC">类别
             </td>
            <td height="30" align="center" valign="middle" bgcolor="#CCCCCC">价格
             </td>
            <td height="30" align="center" valign="middle" bgcolor="#CCCCCC">库存
             </td>
          </tr>
          <% String id=db.hzsb(request.getParameter("bookid")) ;
            String bookname=db.hzsb(request.getParameter("bookname"));
            String author=db.hzsb(request.getParameter("author"));
            String publisher=db.hzsb(request.getParameter("publisher"));
            String category=db.hzsb(request.getParameter("category"));
            ResultSet rs=null;
            String sqls="SELECT * FROM book  WHERE id='"+id+"' OR name='"+
            bookname+"' OR author='"+author+"' OR publisher='"+publisher+"' OR
            type='"+category+"'";
            db.connection();
```

```
            rs=db.query(sqls);
            while (rs.next())
            {   String s1=rs.getString(1);
                String s2=rs.getString(2);
                String s3=rs.getString(3);
                String s4=rs.getString(4);
                String s5=rs.getString(5);
                String s6=rs.getString(6);
                String s7=rs.getString(7);
            %>
            <tr>
              <td height="30" align="center" valign="middle"><%=s1%></td>
              <td height="30" align="center" valign="middle"><%=s2%></td>
              <td height="30" align="center" valign="middle"><%=s3%></td>
              <td height="30" align="center" valign="middle"><%=s4%></td>
              <td height="30" align="center" valign="middle"><%=s5%></td>
              <td height="30" align="center" valign="middle"><%=s6%></td>
              <td height="30" align="center" valign="middle"><%=s7%></td>
            </tr><%   } %>
          </table></form>
        <br><a href="chaz.jsp">返回所有图书首页</a>
      </center>
    </body>
</html>
```

实战演练 3——留言板

【题目】

留言板要求浏览的用户能够留言，并且能够看到全部留言，留言包括留言主题、留言人、留言内容、留言时间，并且要按照最近的留言最先看到的顺序显示留言。

【分析】

(1) 打开 SQL 软件的企业管理器，新建数据库 message，如图 5-58 所示。

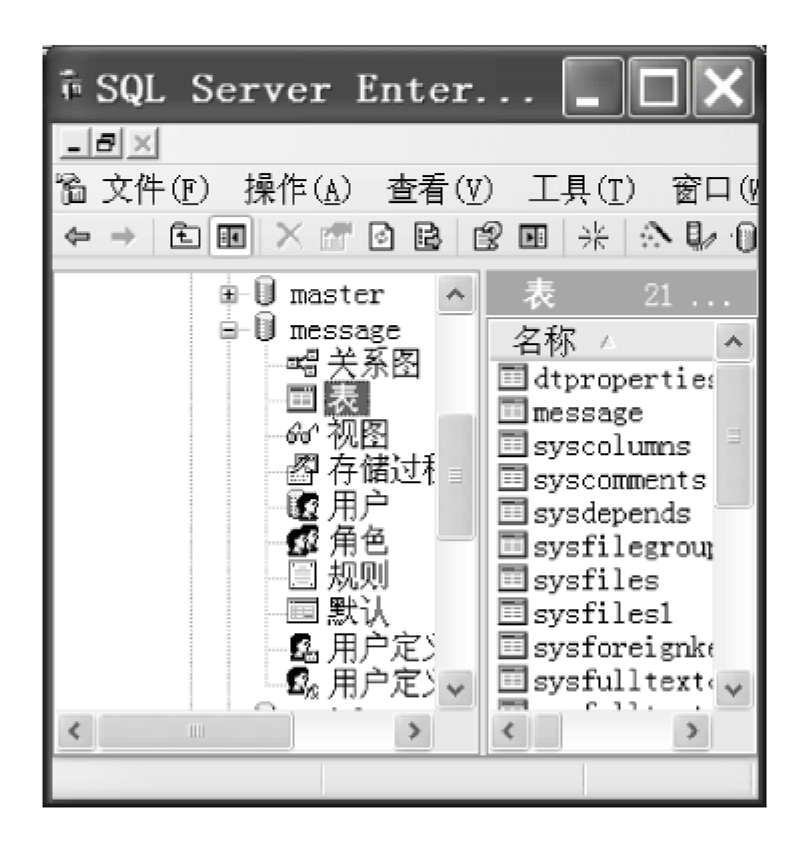


图 5-58 message 数据库

(2) 在 message 数据库中新建库存表 message，表的设计情况如图 5-59 所示。

列名	数据类型	长度	允许空	备注
title	varchar	100		留言主题
name	varchar	20		留言人
message	varchar	200		留言内容
[time]	varchar	20		留言时间

图 5-59　message 表设计

(3) 建立 ODBC 数据源，打开系统管理工具——ODBC 数据源对话框，添加 message 数据源，如图 5-60 所示。



图 5-60　添加 ODBC 数据源

(4) liuyan_show.jsp 文件在这个页面中完成对 message 表中所有信息的倒序显示工作。本页面还包含了"我要留言"的链接，可以链接到留言页面。liuyan.jsp 文件完成的功能是添加记录的界面设计，其中有一个表单和一个表格，当用户想要留言，单击【留言】按钮的时候，表单中的内容就会提交到 liuyan_add.jsp 代码页中。liuyan_add.jsp 文件没有界面设计，完成的功能是接收 liuyan.jsp 文件的表单提交的信息，进行数据库更新，如果更新成功则返回首页查看结果。

【运行效果】

打开开发工具 MyEclipse，在工程 jsp-jdbc 的 bean 文件夹中复制 mydb.java 文件为 mydb_m.java，并修改 mydb_m.java 文件；在这个工程的 WebRoot 文件夹下，新建文件 liuyan_show.jsp、liuyan.jsp、liuyan_add.jsp。程序输入完毕并且保存之后，运行开发工具 MyEclipse 中的 Tomcat 服务器。最后在浏览器中输入地址 http://127.0.0.1:8080/jsp-jdbc/liuyan_show.jsp，就可以看到运行结果了(如图 5-61 和图 5-62 所示)。

【源代码】

【mydb_m.java】

```
package bean;
import java.sql.*;
public class mydb_m {
...
public void connection() {
```



图 5-61 liuyan_show.jsp

图 5-62 liuyan.jsp

```
6        try{
7            Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
8   conn=DriverManager.getConnection("jdbc:odbc: message ");
9   stmt=conn.createStatement(ResultSet.TYPE_SCROLL_SENSITIVE, ResultSet.CONCUR_
    READ_ONLY);
10  …
```

mydb_m.java 补充说明

这里忽略了该文件与上一个题目的文件内容重复的地方，只给大家展示不同之处。

第 3 行，公共类的名字 mydb_m，与本文件名相同；第 8 行，把数据库的名字改成了 message；第 9 行，在 stmt 赋初值的时候不同，stmt 的这个改变是用于留言板记录显示时发挥作用，这其中 TYPE_SCROLL_SENSITIVE 该常量指示可滚动并且通常受其他的更改影响的 ResultSet 对象的类型，CONCUR_READ_ONLY 该常量指示不可以更新的

ResultSet 对象的并发模式。

【源代码】

【liuyan_show.jsp】

```
<%@ page contentType="text/html; charset=gb2312" import="java.sql.*" %>
<html>
  <head>
    <title>显示留言</title>
    <style type="text/css">
      <!--
        .STYLE1 {color: #669900}
      -->
    </style>
  </head>
  <jsp:useBean id="db" class="bean.mydb_m" scope="page"/>
  <body>
    <p align="center"><h2 align="center"><a href="liuyan.jsp" >我  要  留  言</a></h2>
    <div align="center">
      <%
        ResultSet rs=null;
        db.connection();
        String query="select * from message ";
        rs=db.query(query);
        rs.afterLast();
        while (rs.previous())
        { String s1=rs.getString(1);
          String s2=rs.getString(2);
          String s3=rs.getString(3);
          String s4=rs.getString(4);
      %>
      <table width="563" height="118" border="0" bgcolor="#CC9900" cellspacing="2" cellpadding="8">
        <tr>
          <td width="20%" bgcolor="#FFFFCC"><span class="STYLE3 STYLE1">留言主题:</span></td>
          <td colspan="3"><%=s1%></td>
        </tr>
        <tr>
          <td bgcolor="#FFFFCC"><span class="STYLE3 STYLE1">留言人:</span></td>
          <td bgcolor="#ffffff"><%=s2%></td>
        </tr>
        <tr>
          <td bgcolor="#FFFFCC"><p class="STYLE1">留言时间:</p></td>
          <td colspan="3" bgcolor="#ffffff"><%=s4%></td>
        </tr>
        <tr>
          <td bgcolor="#FFFFCC"><p class="STYLE3 STYLE1">留言内容:</p>
```

```
             </td>
42           <td colspan="3" bgcolor="#ffffff"><%=s3%></td>
43         </tr>
44       </table>
45       <p>
46       <%  }  %>
47     </div>
48     <p align="center"><h2 align="center"><a href="liuyan.jsp" >我  要
        留  言</a></h2>
49   </body>
50 </html>
```

liuyan_show.jsp 补充说明

第 1 行,引入 java 类的 SQL 包,因为下面代码用到了 sql 命令。

第 11 行,使用 useBean 动作,使用的类是 bean 文件夹下的 mydb_m 类,类的对象为 db。

第 19 行,db 对象调用 query()方法执行查找 message 表的语句,然后把记录集的返回值赋值给 rs。

第 20 行,为了逆序输出记录,需将 rs 的记录指针移动到最后一行记录之后。

第 21 行到第 46 行,是一个 while 循环,循环条件是当 rs 移动到第一条记录之前就终止循环;第 22 行到第 25 行,把记录集里面的目录信息使用 get 方法取出来赋值给字符串型变量 s1、s2、s3、s4;第 27 行到第 44 行,是一个表格,第 28 行到第 31 行是表格的第一行,在这行的第二个单元格中放置 s1;第 32 行到第 35 行是表格的第二行,在这行的第二个单元格中放置 s2;第 36 行到第 39 行是表格的第三行,在这行的第二个单元格中放置 s4;第 40 行到第 43 行是表格的第四行,在这行的第二个单元格中放置 s3。

第 48 行是“我要留言”的链接 liuyan.jsp。

【源代码】:

【liuyan.jsp】

```
<%@page contentType="text/html; charset=gb2312" import="java.sql.*" %>
<html>
  <head>
    <title>留言板</title>
    <style type="text/css">
      <!--
        .STYLE1 { color: #669900}
      -->
    </style>
  </head>
  <body>
    <form id="form1" name="form1" method="post" action="liuyan_add.jsp">
      <p><h1 align="center" class="STYLE1">留   言   板
      </h1>
       <table width="300" height="123" border="0" align="center" cellpadding="8"
       cellspacing="2" bgcolor="#CC9900">
```

```
        <tr>
          <td width="30%" bgcolor="#FFFFCC"><span class="STYLE1">留言人:
          </span></td>
          <td width="315" bgcolor="#FFFFCC"><input name="name" type="text"
          size="20" maxlength="10"/></td>
        </tr>
        <tr>
          <td bgcolor="#FFFFCC"><span class="STYLE1">留言主题:</span></td>
          <td bgcolor="#FFFFCC"><input name="title" type="text" size="20"
          maxlength="20"/></td>
        </tr>
        <tr>
          <td colspan="2" bgcolor="#FFFFCC"><span class="STYLE1">留言内容:
          </span></td>
        </tr>
        <tr>
          <td colspan="2"><textarea name="message" cols="50" rows="6">
          </textarea></td>
        </tr>
        <tr>
          <td colspan="2" align="center">
            <input name="ok" type="submit" class="STYLE1" value="留言" />
                
            <input name="reset" type="reset" class="STYLE1" value="重填" /></td>
        </tr></table></form>
  </body>
</html>
```

liuyan.jsp 补充说明

在这段代码中 GregorianCalendar 是 Calendar 类(标准的日历)子类，使用 GregorianCalendar 类声明的对象可以调用 Calendar 类的方法获取有关日期和时间的信息。

```
<%@page contentType="text/html; charset=gb2312" import="java.sql.*" %>
<%@page import="java.util.*"%>
<html>
  <head><title>添加留言</title></head>
  <jsp:useBean id="db" class="bean.mydb_m" scope="page"/>
  <body>
    <%String title=db.hzsb(request.getParameter("title"));
      String name=db.hzsb(request.getParameter("name"));
      String message=db.hzsb(request.getParameter("message"));
      GregorianCalendar calen=new GregorianCalendar();
      String t=Integer.toString(calen.get(Calendar.YEAR))+"-"+Integer.toString(calen.
      get(Calendar.MONTH)+1)+"-"+Integer.toString(calen.get(Calendar.DAY_OF_
      MONTH));
      String tm=t+" "+Integer.toString(calen.get(Calendar.HOUR_OF_DAY))+":"+
      Integer.toString(calen.get(Calendar.MINUTE))+":"+Integer.toString(calen.get
      (Calendar.SECOND));
      String time=tm;
```

```
        if (title!=null && name!=null && message!=null)
        {   String rssql="INSERT INTO message VALUES("+"\'"+title+"\'"+","+"\'"
                +name+"\'"+","+"\'"+message+"\'"+","+"\'"+time+"\'"+")";
        db.connection();
        int rsn=db.update(rssql);
        if (rsn>0)
            response.sendRedirect("liuyan_show.jsp");
        }%>
    </body>
</html>
```

拓展——分页功能

【题目】

在上一题的基础上，要求显示留言的时候，每页只显示两条留言记录，计算出数据库中的留言总共可以分成几页显示，并且把能够显示的页面总数和每一页的链接显示出来。另外要求对于留言内容可以识别空格和回车。

【运行效果】

运行效果如图 5-63 所示。

我 要 留 言

留言主题:	zz
留言人:	zzz
留言时间:	2014-5-30 13:23:40
留言内容:	zz

留言主题:	两个苹果
留言人:	sq
留言时间:	2014-5-30 14:33:12
留言内容:	ssssssssssssssssss s s s s s sssssss sssssss s s sssss sssss s s sss sss s ssssssssssssssssss

[1] [2] [3] [4] [5] [6] [7] [8] 共8页

图 5-63 显示留言结果界面 liuyan_show.jsp

【源代码】

【replace.java】

```
package bean;
public class replace {
  private String content="";
  public void setContent(String content)
  {
      this.content=content;
      this.content=this.content.replaceAll(" ", " ");
```

```
8          this.content=this.content.replaceAll("\r\n", "<br>");
9      }
10     public String getContent()
11     {return this.content;}
12 }
```

replace.java 补充说明

第 7 行，把留言中的空格“ ”用 表示；第 8 行，把留言中的回车用“
”表示。

【源代码】

【liuyan_show.jsp】

```
<%@ page contentType="text/html; charset=gb2312" language="java" import="java.sql.*" %>
<html>
  <head>
    <title>显示留言</title>
    <style type="text/css">
      .STYLE1 {color: #669900}
    </style>
  </head>
  <jsp:useBean id="db" class="bean.mydb" scope="page"/>
  <jsp:useBean id="zh" class="bean.replace" scope="page"/>
  <body>
    <p align="center"><h2 align="center"><a href="liuyan.jsp" >我  要  留  言</a></h2>
    <div align="center">
      <%
        int row_page=0;
        int per_show=2;
        int row_count=0;
        int now_page=1;
          ResultSet rs=null;
          db.connection();
          String query="select * from message ";
          rs=db.query(query);
        rs.last();
          row_count=rs.getRow();
        row_page=(row_count%per_show==0)?(row_count/per_show):(row_count/per_show+1);
        String temp=request.getParameter("to_page");
        if(temp==null)
          now_page=1;
        else
          now_page=Integer.parseInt(temp);
        if(row_count>0)
        {
          rs.absolute(per_show*(now_page-1)+1);
          for(int i=1;i<=per_show;i++)
```

```
35              {
36                if(!rs.isAfterLast())
37                {
38                  String s1=rs.getString(1);
39                  String s2=rs.getString(2);
40                  String s3=rs.getString(3);
41                  String s4=rs.getString(4);
42                  zh.setContent(s3);
43                  s3=zh.getContent();
44        %>
45        <table width="563" height="118" border="0" bgcolor="#CC9900" cellspacing=
          "2" cellpadding="8">
46          <tr>
47            <td width="20%" bgcolor="#FFFFCC"><span class="STYLE3 STYLE1">留
            言主题:</span></td>
48            <td colspan="3"><%=s1%></td>
49          </tr>
50          <tr>
51             <td bgcolor="#FFFFCC"><span class="STYLE3 STYLE1">留言人:
             </span></td>
52            <td bgcolor="#ffffff"><%=s2%></td>
53          </tr>
54          <tr>
55            <td bgcolor="#FFFFCC"><p class="STYLE1">留言时间:</p>        </td>
56            <td colspan="3" bgcolor="#ffffff"><%=s4%></td>
57          </tr>
58          <tr>
59            <td bgcolor="#FFFFCC"><p class="STYLE3 STYLE1">留言内容:</p>
            </td>
60            <td colspan="3" bgcolor="#ffffff"><%=s3%></td>
61          </tr>
62        </table>
63        <p>
64        <%
65          rs.next();}}}
66        %>
67      </div>
68      <p align="right">
69      <%  for(int i=1;i<=row_page;i++)
70        {
71      %>
72       <a href="shici.jsp?to_page=<%=i %>" >[<%=i %>]</a> 
        <%  }  %>
73      共<%=row_page %>页
74    </body>
75  </html>
```

liuyan_show.jsp 补充说明

第 15 到第 18 行,row_page 记录集总共可以显示几页;per_show 设置每页显示几条记录; row_count 记录集总数;now_page 当前页码。

第 24 行，整形变量 row_count 赋值为记录集总数。
第 25 行，计算记录集总共可以显示几页。
第 26 行，通过参数获取当前页码。
第 33 行，将指针移到需要显示页的第 1 个数据。
第 42 行，把留言中的空格和回车做相应的转换才能在输出中显示出来。

本章小结

JSP 数据库应用开发是 JSP 开发中的重点和难点，只有掌握了数据库的开发，JSP 才能发挥其最大功效。本章从数据库出发，依次介绍了 SQL 数据库的安装与使用、JDBC 驱动程序、JDBC 编程，并且通过若干实例介绍了如何在 JSP 中进行数据库应用开发。

本章习题

综合练习 1——教育网站

【题目】

制作一个教育网站(图 5-64)，要求登录网站的用户可以浏览网站的所有内容，并且可以注册成为网站的会员，成为会员之后就可以下载网站上的学习资料。浏览网站的用户可以在网站上交流，所以要求网站有一个留言板模块(图 5-65 和图 5-66)。



图 5-64　网站首页



图 5-65 可以下载学习资料的页面 1



图 5-66 可以下载学习资料的页面 2

综合练习 2——购物网站

【题目】

制作一个购物网站，要求网站的系统分为两个部分：一部分是管理员的后台管理；另一部分是用户的前台操作。管理员可以对商品的库存进行管理(添加、修改、删除、查询商品)，还可以查看会员的购物订单，另外还可以对网站的会员进行查看或删除。普通用户浏览这

个购物网站的时候可以查看网站商品的库存情况，如果想要购买的话，需要先注册成为网站的会员，然后才可以把商品放到购物车中，确定购买之后可以提交购物车中的商品形成订单。

【分析】

购物网站制作要求及内容如图 5-67 所示。

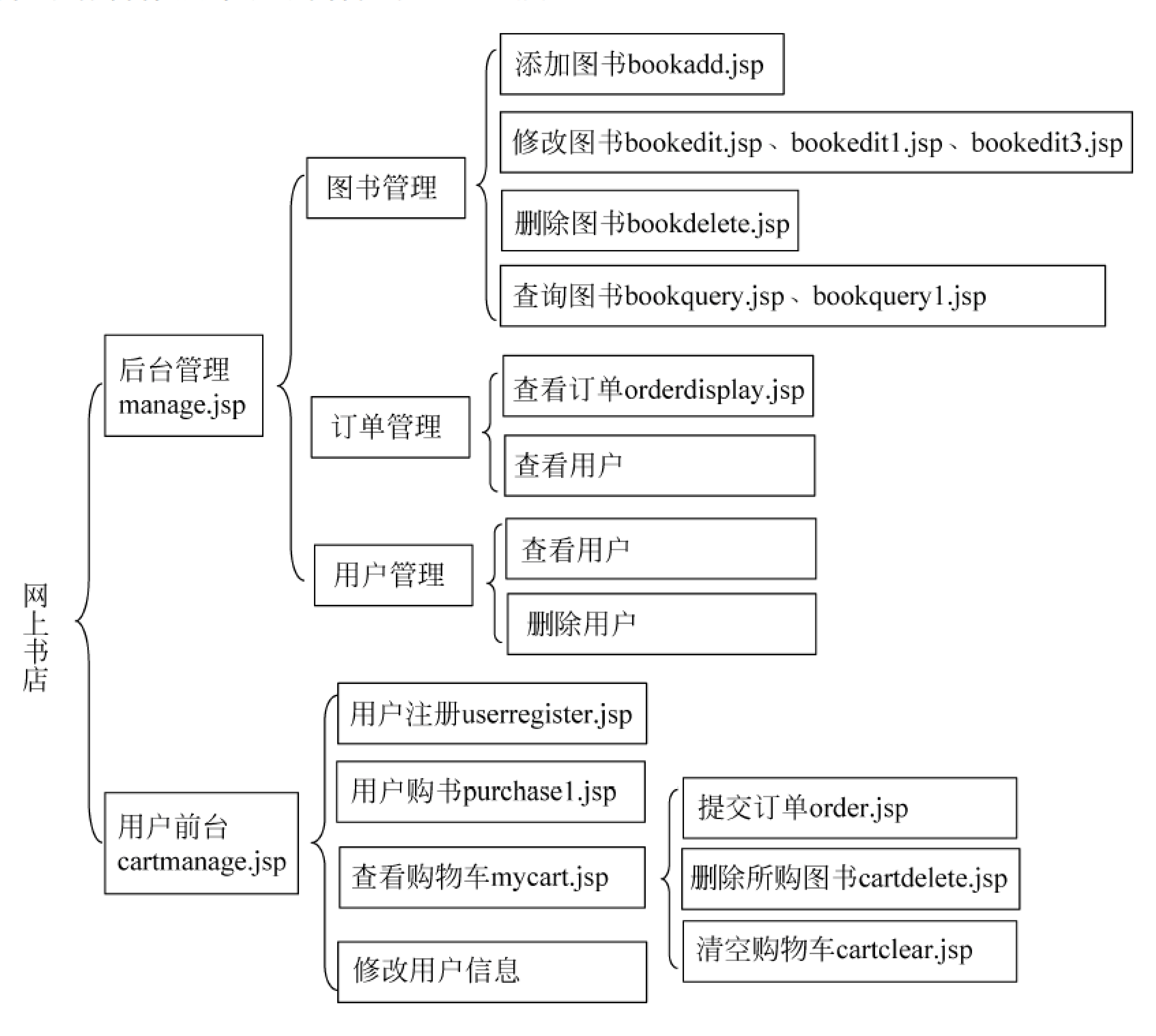


图 5-67　购物网站制作要求及内容

（1）管理员模板 mb.dwt.jsp，管理员页面 manage.jsp 文件可以显示 book 数据表的所有信息，如图 5-68 所示。

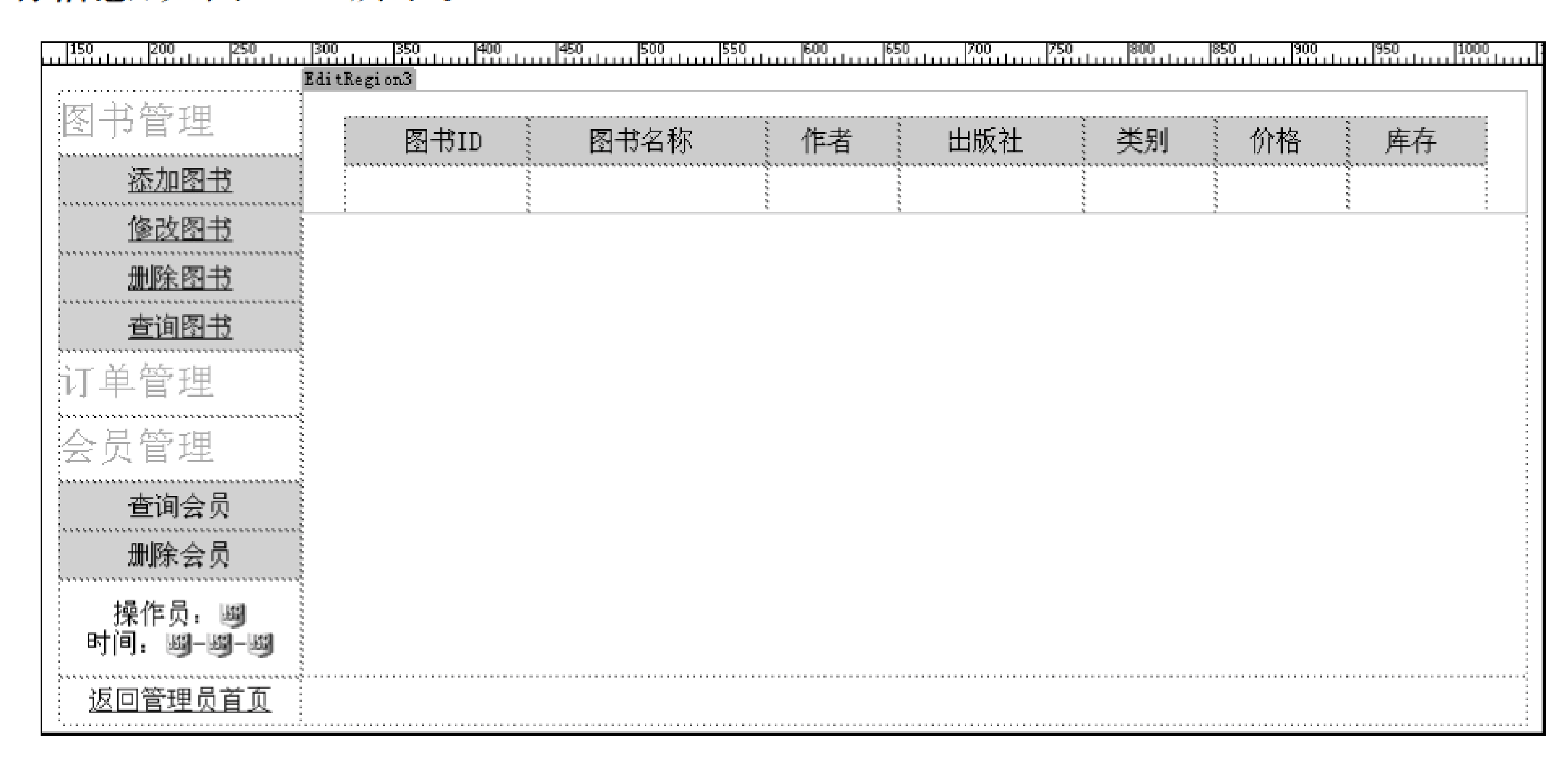


图 5-68　管理员界面设计模板

（2）用户模板 cart.dwt.jsp，会员登录以后 cartmanage.jsp 页面可以显示会员能够购买的所有书籍，如图 5-69 所示。

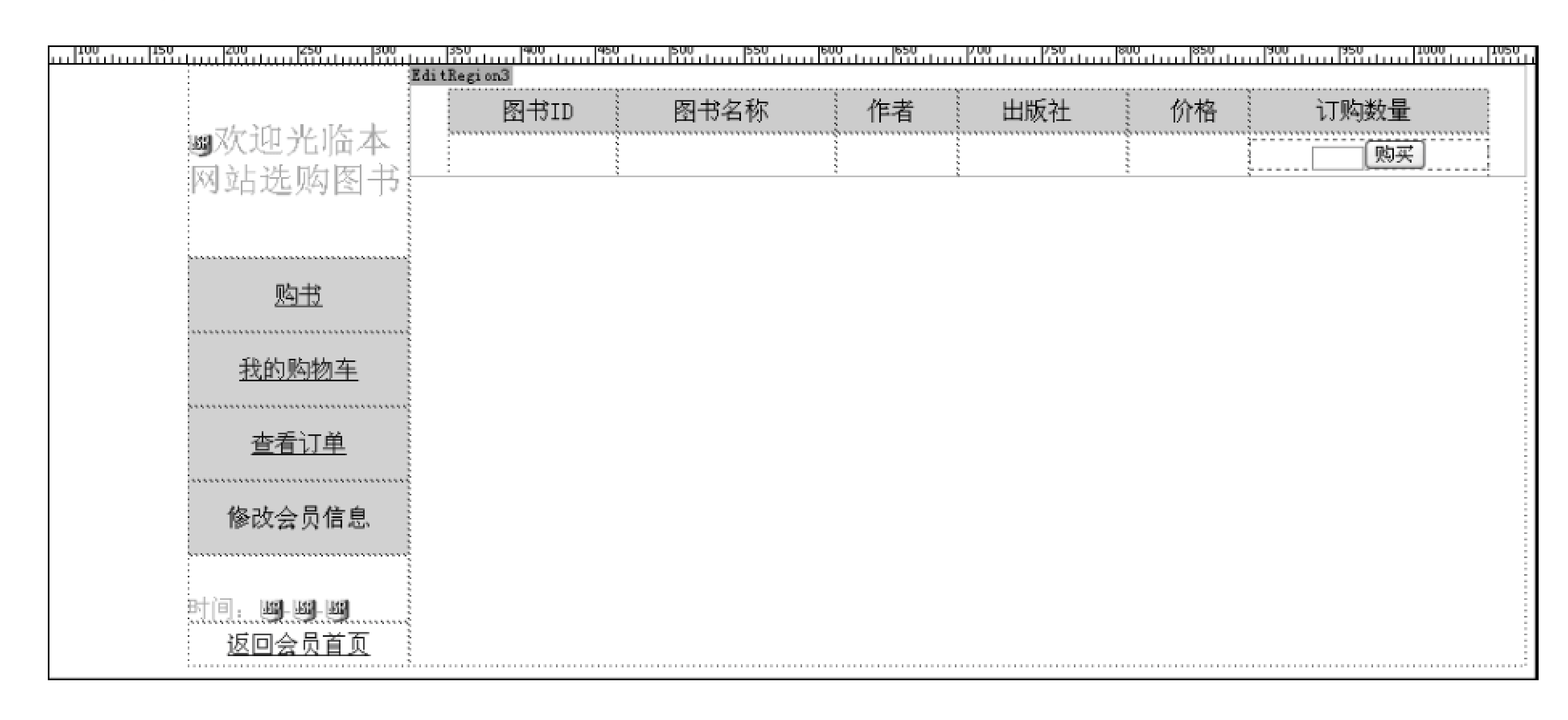


图 5-69　用户界面设计模板

（3）建立数据库，如图 5-70～图 5-74 所示。

设计表“admin”，位置是“bc

列名	数据类型	长度	允许空
id	varchar	10	
password	varchar	10	

图 5-70　管理员表 admin

设计表“book”，位置是“boc

列名	数据类型	长度	允许空
id	nvarchar	50	
name	nvarchar	50	
author	nvarchar	50	
publisher	nvarchar	50	
type	nvarchar	50	
price	nvarchar	50	
stock	nvarchar	50	

图 5-71　书籍库存表 book

设计表“customer”，位置是

列名	数据类型	长度	允许空
id	varchar	10	
password	varchar	15	
name	varchar	15	
sex	varchar	8	
address	varchar	50	
code	varchar	15	
tel	varchar	20	
email	varchar	30	

图 5-72　会员注册表

设计表“userorder”，位置是

列名	数据类型	长度	允许空
id	varchar	10	
username	varchar	10	
[day]	varchar	20	
money	varchar	20	

图 5-73　会员订单表 userorder

设计表“orderlist”，位置是

列名	数据类型	长度	允许空
id	varchar	10	
[user]	varchar	20	
book	varchar	30	
[sum]	varchar	10	
money	varchar	20	

图 5-74　订单详细信息表

(4) 数据库表之间的关系,如图 5-75 和图 5-76 所示。



图 5-75　表 orderlist 与表 userorder 的关系

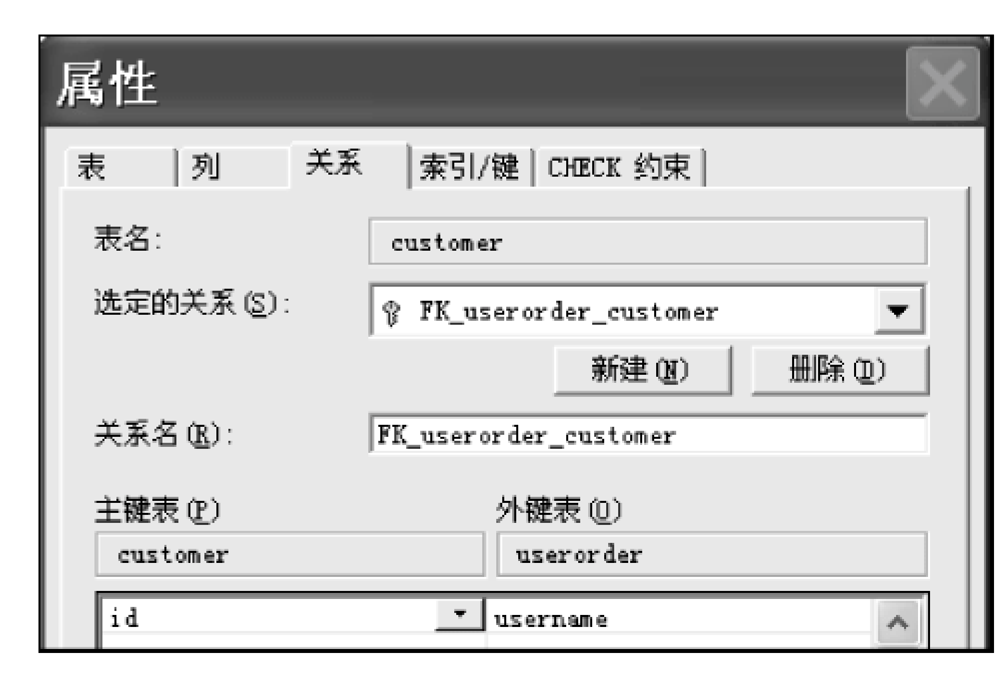


图 5-76　表 customer 与表 userorder 的关系

【运行效果】

(1) 登录界面(图 5-77)index.html,选择管理员或用户的身份,然后输入相应的用户名和密码,管理员进入 manage.jsp 页面,用户进入 cartmanage.jsp 页面。

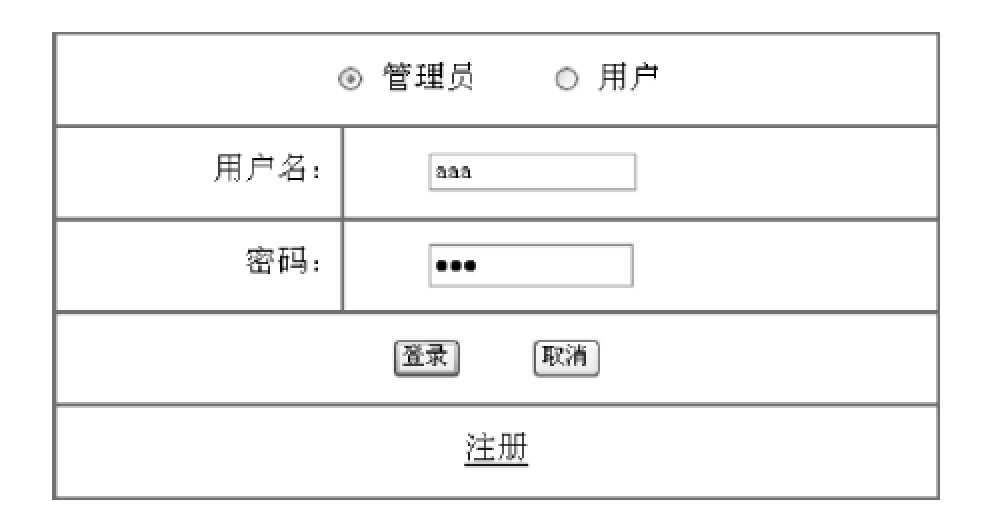


图 5-77　登录界面

(2) 管理员界面(图 5-78)manage.jsp 显示 book 数据表的所有信息。



图书ID	图书名称	作者	出版社	类别	价格	库存
1	计算机基础	xx	高教出版社	科技	20	0
2	财富	gg	上海	科技	20	6
3	旅行者	ss	上海	科技	15	10
4	绝对小孩儿	nn	文艺出版社	科技	23	8
5	网络技术	ww	人民邮电出版社	科技	12	12
6	瑜伽	tt	人民邮电出版社	科技	34	6
7	广告创意	ww	机械工业出版社	科技	44	0
8	JSP	JSP	JSP	科技	30	0

图 5-78　管理员界面 manage.jsp

(3) 管理员——修改图书(图 5-79 和图 5-80)bookedit.jsp,bookedit1.jsp。



图 5-79　修改图书 1



图 5-80　修改图书 2

(4) 管理员——删除图书(图 5-81)bookdelete.jsp，bookdelete1.jsp。

(5) 管理员——图书查询(图 5-82 和图 5-83)bookquery.jsp，bookquery1.jsp。

(6) 管理员——对订单的查看、删除核对会员的查看删除功能与图书的查看和删除的功能实现。

(7) 会员注册(userregister.jsp 页面),注册为会员(图 5-84)以后就能够从登录页面 index.html 进入到会员首页(图 5-85)。



图 5-81　删除图书

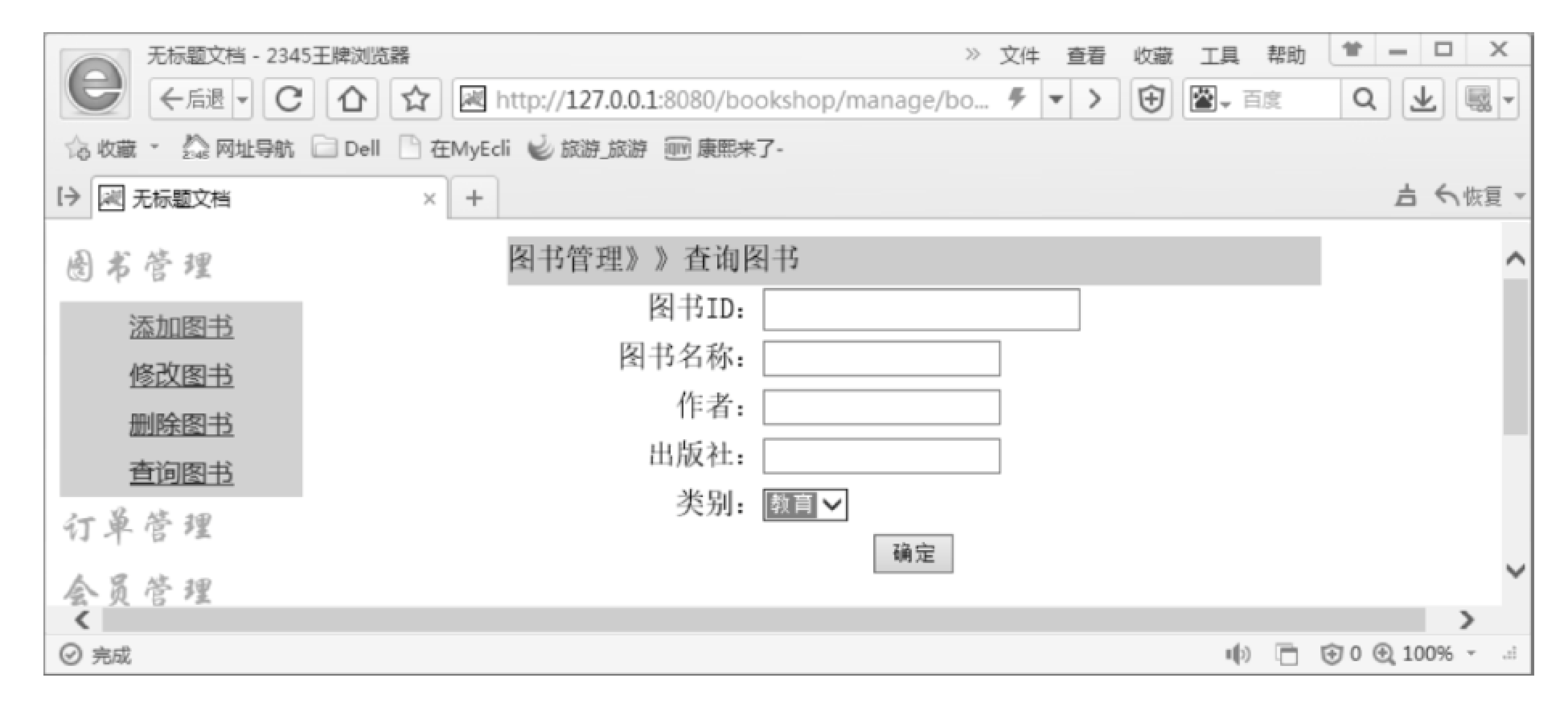


图 5-82　查询图书 1

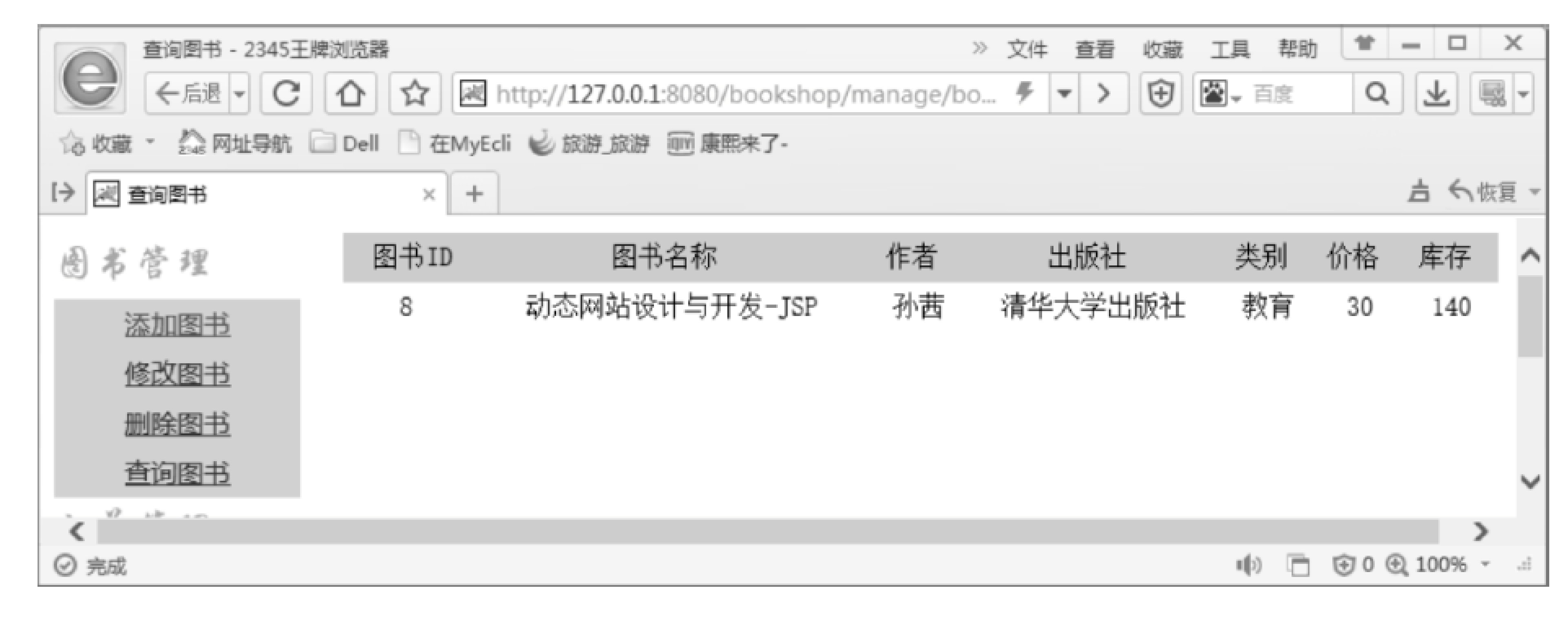


图 5-83　查询图书 2



图 5-84　会员注册 userregister.jsp



图书ID	图书名称	作者	出版社	库存	价格	订购数量
1	计算机基础	xx	高教出版社	0	20	购买
2	财富	gg	上海	6	20	购买
3	旅行者	ss	上海	10	15	购买
4	绝对小孩儿	nn	文艺出版社	8	23	购买
5	网络技术	ww	人民邮电出版社	12	12	购买
6	瑜伽	tt	人民邮电出版社	6	34	购买
7	广告创意	ww	机械工业出版社	0	44	购买
8	动态网站设计与开发-JSP	孙茜	清华大学出版社	140	30	购买

图 5-85　用户购买首页

(8) 在用户首页 cartmanage.jsp 填写购书的数量，并单击“购买”按钮，执行 parchase.jsp 代码页，然后跳转到 mycart.jsp 页面。

(9) 购物车 mycart.jsp 页面(图 5-86)显示要买的书籍，如果放弃可以单击“清空购物车 cartclear.jsp”或者“我不要了 cartdelete.jsp”；如果确定不再改动单击“提交订单”执行 order.jsp，然后跳转到订单页面 rderdisplay.jsp。

(10) 查看订单页面(图 5-87)orderdisplay.jsp，单击“查看”可以跳转到 orderdetail.jsp 页面查看订单详细信息。

(11) 订单详细信息页面(图 5-88)orderdetail.jsp。

源代码见附录1。



图 5-86　购物车



图 5-87　查看订单



图 5-88　订单详细信息

附录 1 源代码

Appendtx 1

【mydb. java】

本文件与本章前面例题中的代码相同，只是在连接数据库的语句上做了修改，源代码如附图 1-1 所示。

【cart. java】

```
package bean;
public class cart {
    private String bookId;
    private int num;
    public cart(){
        bookId="";
        num=0;
    }
    public cart(String bookId,int num){
        this.bookId=bookId;
        this.num=num;
    }
    public String getBookId(){
        return(bookId);
    }
    public void setBookId(String bookId){
        this.bookId=bookId;
    }
    public int getNum(){
        return(num);
    }
    public void setNum(int num){
        this.num=num;
    }
}
```

```
bookshop
  src
    bean
      cart.java
      mydb.java
  JRE System Library [MyE
  J2EE 1.4 Libraries
  WebRoot
    cart
      cartclear.jsp
      cartdelete.jsp
      cartmanage.jsp
      mycart.jsp
      order.jsp
      orderdetail.jsp
      orderdisplay.jsp
      purchase.jsp
      userregister.jsp
    manage
      bookadd.jsp
      bookdelete.jsp
      bookdelete1.jsp
      bookedit.jsp
      bookedit1.jsp
      bookquery.jsp
      bookquery1.jsp
      manage.jsp
    META-INF
    WEB-INF
    index.html
```

附图 1-1 源代码

【index. html】

```
<html >
  <head>
    <title>书店</title>
    <style type="text/css">
      <!--
        .STYLE9 {font-family: Arial, Helvetica, sans-serif; font-size: 18px; }
```

```
    -->
  </style>
</head>
<body>
  <form id="form5" name="form5" method="post" action="login.jsp">
    <table width="500" border="0" align="center" cellpadding="0" cellspacing="2"
    bgcolor="#0099CC">
      <tr>
          <td height="50" colspan="2" align="center" valign="middle" bgcolor=
          "#FFFFFF"><label>
           <input name="role" type="radio" id="radio" value="manager" checked="checked" />
           <span class="STYLE9">管理员</span>    
           <input type="radio" name="role" id="radio2" value="user" />
           <span class="STYLE9">用户</span></label>
        </td>
      </tr>
    <tr>
      <td width="203" align="right" valign="middle" bgcolor="#FFFFFF"><span
      class="STYLE9">用户名:</span></td>
      <td width="394" height="50" valign="middle" bgcolor="#FFFFFF"><label>
           
        <input name="username" type="text" id="textfield" size="15" />
        </label>
      </td>
    </tr>
    <tr>
      <td align="right" valign="middle" bgcolor="#FFFFFF"><span class="STYLE9">
      密码:</span></td>
      <td height="50" valign="middle" bgcolor="#FFFFFF"><label>   
         
        <input name="password" type="password" id="textfield2" size="15" />
        </label>
      </td>
    </tr>
    <tr>
      <td height="50" colspan="2" align="center" valign="middle" bgcolor="#FFFFFF">
      <label>
        <input type="submit" name="button" id="button" value="登录" />
        </label>
        <label>    
        <input type="reset" name="button2" id="button2" value="取消" />
        </label>
      </td>
    </tr>
    <tr>
      <td height="50" colspan="2" align="center" valign="middle" bgcolor="#FFFFFF"
      class="STYLE9"><a href="cart/userregister.jsp">注册</a></td>
    </tr>
  </table>
```

```
  </form>
 </body>
</html>
```

【login. jsp】

```
<%@ page contentType="text/html; charset=gb2312" language="java" import="java.sql.*" %>
<html>
  <body>
  <jsp:useBean id="db" class="bean.mydb" scope="page"/>
    <%
      String username=request.getParameter("username");
      String password=request.getParameter("password");
      String role=request.getParameter("role");
      ResultSet rs=null;
      //如果是管理员页面跳转到管理页面
      if (role.equals("manager"))
      {
        db.connection();
        rs=db.query("SELECT * FROM admin WHERE id="+"\'"+username+"\'"+
        "AND password="+"\'"+password+"\'");
        while(rs.next())
          {
            session.setAttribute("username",username);
            response.sendRedirect("manage/manage.jsp");
          }
        }
        //如果是用户页面跳转到用户页面
        if(role.equals("user"))
        {
          db.connection();
          rs=db.query("SELECT * FROM customer WHERE id="+"\'"+username+"\'"
          + "AND password="+"\'"+password+"\'");
          while(rs.next())
          {
            session.setAttribute("username",username);
            response.sendRedirect("cart/cartmanage.jsp");
          }
        }
    %>
  </body>
</html>
```

【manage. jsp】

```
<%@ page contentType="text/html; charset=gb2312" language="java" import="java.sql.*,
java.util.*" errorPage="" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.
w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"><!-- InstanceBegin template=
```

```
"/Templates/mb.dwt.jsp" codeOutsideHTMLIsLocked="false" -->
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=gb2312" />
    <!-- InstanceBeginEditable name="doctitle" -->
    <title>管理员</title>
    <!-- InstanceEndEditable -->
    <!-- InstanceBeginEditable name="head" -->
    <style type="text/css">
      <!--
        .STYLE14 {color: #FF6600}
        .STYLE18 {font-size: 24}
      -->
    </style>
    <!-- InstanceEndEditable -->
    <style type="text/css">
      <!--
        .STYLE10 {font-family: "宋体"}
        .STYLE12 {font-family: Arial, Helvetica, sans-serif; font-size: 16px; }
        .STYLE2 {font-family: "华文行楷";
        font-size: 24px;
        color: #FF9900;
      }
      -->
    </style>
  </head>
  <jsp:useBean id="db" class="bean.mydb" scope="page"/>
  <body>
    <table width="900" border="0" align="center" cellpadding="0" cellspacing="0">
      <tr>
        <td width="149" height="40"><span class="STYLE2">图书管理</span></td>
          <td width="751" rowspan="10" align="center" valign="top"><!--
          InstanceBeginEditable name="EditRegion3" -->
          <form id="form1" name="form1" method="post" action="">
            <table width="700" border="0" cellspacing="0" cellpadding="0">
              <tr>
                  <td height="30" align="center" valign="middle" bgcolor="#CCCCCC">
                  <span class="STYLE10 STYLE18"> 图书 ID</span></td>
                  <td height="30" align="center" valign="middle" bgcolor="#CCCCCC">
                  <span class="STYLE10 STYLE18">图书名称  </span></td>
                  <td height="30" align="center" valign="middle" bgcolor="#CCCCCC">
                  <span class="STYLE10 STYLE18">作者  </span></td>
                  <td height="30" align="center" valign="middle" bgcolor="#CCCCCC">
                  <span class="STYLE10 STYLE18">出版社  </span></td>
                  <td height="30" align="center" valign="middle" bgcolor="#CCCCCC">
                  <span class="STYLE10 STYLE18">类别  </span></td>
                  <td height="30" align="center" valign="middle" bgcolor="#CCCCCC">
                  <span class="STYLE10 STYLE18">价格  </span></td>
                  <td height="30" align="center" valign="middle" bgcolor="#CCCCCC">
                  <span class="STYLE10 STYLE18">库存  </span></td>
```

```
<td height="30" align="center" valign="middle" bgcolor="#CCCCCC">
<span class="STYLE18"></span></td>
</tr>
<%
ResultSet rs=null;
db.connection();
String query="select * from book";
rs=db.query(query);
while (rs.next())
{ String s1=rs.getString(1);
  String s2=rs.getString(2);
  String s3=rs.getString(3);
  String s4=rs.getString(4);
  String s5=rs.getString(5);
  String s6=rs.getString(6);
  String s7=rs.getString(7);
%>
<tr>
  <td height="30" align="center" valign="middle"><span class=
  "STYLE18"><%=s1%></span></td>
  <td height="30" align="center" valign="middle"><span class=
  "STYLE18"><%=s2%></span></td>
  <td height="30" align="center" valign="middle"><span class=
  "STYLE18"><%=s3%></span></td>
  <td height="30" align="center" valign="middle"><span class=
  "STYLE18"><%=s4%></span></td>
  <td height="30" align="center" valign="middle"><span class=
  "STYLE18"><%=s5%></span></td>
  <td height="30" align="center" valign="middle"><span class=
  "STYLE18"><%=s6%></span></td>
  <td height="30" align="center" valign="middle"><span class=
  "STYLE18"><%=s7%></span></td>
</tr><% } %>
</table>
</form>
<!-- InstanceEndEditable --></td>
</tr>
<tr>
<td height="30" align="center" bgcolor="#FFCC66"><span class="STYLE12">
<a href="bookadd.jsp">添加图书</a></span></td>
</tr>
<tr>
<td height="30" align="center" bgcolor="#FFCC66"><span class="STYLE12">
<a href="bookedit.jsp">修改图书</a></span></td>
</tr>
<tr>
<td height="30" align="center" bgcolor="#FFCC66"><span class="STYLE12">
<a href="bookdelete.jsp">删除图书</a></span></td>
</tr>
```

```
        <tr>
          <td height="30" align="center" bgcolor="#FFCC66"><span class="STYLE12">
          <a href="bookquery.jsp">查询图书</a></span></td>
        </tr>
        <tr>
          <td height="40" class="STYLE2">订单管理</td>
        </tr>
        <tr>
          <td height="40" class="STYLE2">会员管理</td>
        </tr>
        <tr>
          <td height="30" align="center" bgcolor="#FFCC66" class="STYLE12">查询会
          员</td>
        </tr>
        <tr>
          <td height="30" align="center" bgcolor="#FFCC66" class="STYLE12">删除会
          员</td>
        </tr>
        <%
          GregorianCalendar d=new GregorianCalendar();
          String year=Integer.toString(d.get(Calendar.YEAR));
          String month=Integer.toString(d.get(Calendar.MONTH)+1);
          String day=Integer.toString(d.get(Calendar.DAY_OF_MONTH));
        %>
        <tr>
          <td height="60" align="center"><span class="STYLE10">操作员:<%=
          (String)session.getAttribute("username")%><br />
            时间:<%=year%>-<%=month%>-<%=day%></span></td>
        </tr>
        <tr>
          <td height="30" align="center"><a href="manage.jsp">返回管理员首页</a>
          </td>
        </tr>
      </table>
    </body>
  <!-- InstanceEnd --></html>
```

【bookadd.jsp】

```
<%@ page contentType="text/html; charset=gb2312" language="java" import="java.sql.*,
java.util.*" errorPage="" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.
w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"><!-- InstanceBegin template=
"/Templates/mb.dwt.jsp" codeOutsideHTMLIsLocked="false" -->
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312" />
  <!-- InstanceBeginEditable name="doctitle" -->
  <title>无标题文档</title>
  <!-- InstanceEndEditable -->
```

```
<!-- InstanceBeginEditable name="head" -->
<style type="text/css">
 <!--
  .STYLE19 {font-family: "宋体"; font-size: 16; }
  .STYLE19 {font-family: Arial, Helvetica, sans-serif; font-size: 16px; }
 -->
</style>
<!-- InstanceEndEditable -->
<style type="text/css">
 <!--
  .STYLE14 {color: #FF6600}
  .STYLE18 {font-size: 24}
 -->
</style>
<style type="text/css">
 <!--
  .STYLE10 {font-family: "宋体"}
  .STYLE12 {font-family: Arial, Helvetica, sans-serif; font-size: 16px; }
  .STYLE2 {font-family: "华文行楷";
    font-size: 24px;
    color: #FF9900;
    }
 -->
</style>
</head>
<jsp:useBean id="db" class="bean.mydb" scope="page"/>
<body>
  <table width="900" border="0" align="center" cellpadding="0" cellspacing="0">
   <tr>
    <td width="149" height="40"><span class="STYLE2">图书管理</span></td>
     <td width="751" rowspan="10" align="center" valign="top"><!--
     InstanceBeginEditable name="EditRegion1" -->
     <form id="form1" name="form1" method="post" action="">
      <table width="500" border="0" align="center" cellpadding="0" cellspacing="0">
        <tr>
            <td height="30" colspan="2" align="left" valign="middle" bgcolor=
            "#CCCCCC" class="STYLE7"><label><span class="STYLE10">图书管
            理》》添加图书</span></label></td>
        </tr>
        <tr>
          <td height="15" align="right" valign="middle" class="STYLE7 STYLE8">图书
          ID:</td>
          <td height="30" align="left" valign="middle"><input name="bookid" type=
          "text" id="textfield" /></td>
        </tr>
        <tr>
          <td height="30" align="right" valign="middle" class="STYLE7 STYLE8">图书
          名称:</td>
          <td height="30" align="left" valign="middle"><span class="STYLE19">
```

```
        <label>
          <input type="text" name="bookname" id="textfield2" />
        </label>
      </span></td>
    </tr>
    <tr>
      <td height="30" align="right" valign="middle" class="STYLE7 STYLE8">作
      者:</td>
      <td height="30" align="left" valign="middle"><span class="STYLE19">
        <label>
          <input type="text" name="author" id="textfield3" />
        </label>
      </span></td>
    </tr>
    <tr>
      <td height="30" align="right" valign="middle" class="STYLE7 STYLE8">出版
      社:</td>
      <td height="30" align="left" valign="middle"><span class="STYLE19">
        <label>
          <input type="text" name="publisher" id="textfield4" />
        </label>
      </span></td>
    </tr>
    <tr>
      <td height="30" align="right" valign="middle" class="STYLE7 STYLE8">类
      别:</td>
      <td height="30" align="left" valign="middle"><span class="STYLE19">
        <label>
          <select name="category" id="select">
            <option selected="selected">科技</option>
            <option>艺术</option>
            <option>生活</option>
            <option>社会</option>
            <option>娱乐</option>
            <option>教育</option>
          </select>
        </label>
      </span></td>
    </tr>
    <tr>
      <td height="30" align="right" valign="middle" class="STYLE7 STYLE8">价
      格:</td>
      <td height="30" align="left" valign="middle"><span class="STYLE19">
        <label>
          <input type="text" name="price" id="textfield5" />
        </label>
      </span></td>
    </tr>
    <tr>
```

```
        <td height="30" align="right" valign="middle" class="STYLE7 STYLE8">库
        存:</td>
        <td height="30" align="left" valign="middle"><span class="STYLE19">
          <label>
           <input type="text" name="stock" id="textfield6" />
          </label>
        </span></td>
      </tr>
      <tr>
        <td height="30" colspan="2" align="center" valign="middle"><label>
            <input type="submit" name="button" id="button" value="确定" />
          </label>
          <label>
            <input type="reset" name="button2" id="button2" value="重填" />
          </label></td>
      </tr>
      <%
       //获取表单提交数据
       String bookid=db.hzsb(request.getParameter("bookid")) ;
       String bookname=db.hzsb(request.getParameter("bookname")) ;
       String author=db.hzsb(request.getParameter("author"));
       String publisher=db.hzsb(request.getParameter("publisher"));
       String category=db.hzsb(request.getParameter("category"));
       String price=db.hzsb(request.getParameter("price"));
       String stock=db.hzsb(request.getParameter("stock"));

       //如果数据都不为空,插入数据库 book 表中
       if (bookid!=null && bookname!=null && author!=null && publisher!=null &&
       category!=null && price!=null && stock!=null)
       {
         String rssql="insert into book values("+"\'"+bookid+"\'"+","+"\'"+bookname
         +"\'"+","+"\'"+author+"\'"+","+"\'"+publisher+"\'"+","+"\'"+
         category+"\'"+","+"\'"+price+"\'"+","+"\'"+stock+"\'"+")";
         db.connection();
         int rsn=db.update(rssql);
         if (rsn>0)
         response.sendRedirect("manage.jsp");
       }
      %>
    </table>
  </form>
 <!-- InstanceEndEditable --></td>
</tr>
<tr>
 <td height="30" align="center" bgcolor="#FFCC66"><span class="STYLE12">
 <a href="bookadd.jsp">添加图书</a></span></td>
</tr>
<tr>
 <td height="30" align="center" bgcolor="#FFCC66"><span class="STYLE12">
```

```
    <a href="bookedit.jsp">修改图书</a></span></td>
   </tr>
   <tr>
    <td height="30" align="center" bgcolor="#FFCC66"><span class="STYLE12">
    <a href="bookdelete.jsp">删除图书</a></span></td>
   </tr>
   <tr>
    <td height="30" align="center" bgcolor="#FFCC66"><span class="STYLE12">
    <a href="bookquery.jsp">查询图书</a></span></td>
   </tr>
   <tr>
    <td height="40" class="STYLE2">订单管理</td>
   </tr>
   <tr>
    <td height="40" class="STYLE2">会员管理</td>
   </tr>
   <tr>
    <td height="30" align="center" bgcolor="#FFCC66" class="STYLE12">查询会员</td>
   </tr>
   <tr>
    <td height="30" align="center" bgcolor="#FFCC66" class="STYLE12">删除会员</td>
   </tr>
   <%
     GregorianCalendar d=new GregorianCalendar();
     String year=Integer.toString(d.get(Calendar.YEAR));
     String month=Integer.toString(d.get(Calendar.MONTH)+1);
     String day=Integer.toString(d.get(Calendar.DAY_OF_MONTH));
   %>

   <tr>
    <td height="60" align="center"><span class="STYLE10">操作员:<%=(String)
    session.getAttribute("username")%><br />
      时间:<%=year%>-<%=month%>-<%=day%></span></td>
   </tr>
   <tr>
    <td height="30" align="center"><a href="manage.jsp">返回管理员首页</a></td>
   </tr>
  </table>

</body>
<!-- InstanceEnd --></html>
```

【bookdelete.jsp】

```
<%@ page contentType="text/html; charset=gb2312" language="java" import="java.sql.*,
java.util.*" errorPage="" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.
w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"><!-- InstanceBegin template=
"/Templates/mb.dwt.jsp" codeOutsideHTMLIsLocked="false" -->
```

```
<head>
 <meta http-equiv="Content-Type" content="text/html; charset=gb2312" />
  <!-- InstanceBeginEditable name="doctitle" -->
  <title>删除图书</title>
  <!-- InstanceEndEditable -->
  <!-- InstanceBeginEditable name="head" -->
  <style type="text/css">
   <!--
    .STYLE14 {color: #FF6600}
    .STYLE18 {font-size: 24}
   -->
 </style>
 <!-- InstanceEndEditable -->
 <style type="text/css">
  <!--
    .STYLE10 {font-family: "宋体"}
    .STYLE12 {font-family: Arial, Helvetica, sans-serif; font-size: 16px; }
    .STYLE2 {font-family: "华文行楷";
      font-size: 24px;
      color: #FF9900;
      }
  -->
 </style>
</head>
<jsp:useBean id="db" class="bean.mydb" scope="page"/>
<body>
 <table width="900" border="0" align="center" cellpadding="0" cellspacing="0">
  <tr>
   <td width="149" height="40"><span class="STYLE2">图书管理</span></td>
    <td width="751" rowspan="10" align="center" valign="top" ><!--
    InstanceBeginEditable name="EditRegion3" -->
    <form id="form1" name="form1" method="post" action="">
      <table width="700" border="0" cellspacing="0" cellpadding="0">
        <tr>
          <td height="30" align="center" valign="middle" bgcolor="#CCCCCC">
          <span class="STYLE10 STYLE18"> 图书 ID</span></td>
          <td height="30" align="center" valign="middle" bgcolor="#CCCCCC">
          <span class="STYLE10 STYLE18">图书名称  </span></td>
          <td height="30" align="center" valign="middle" bgcolor="#CCCCCC">
          <span class="STYLE10 STYLE18">作者  </span></td>
          <td height="30" align="center" valign="middle" bgcolor="#CCCCCC">
          <span class="STYLE10 STYLE18">出版社  </span></td>
          <td height="30" align="center" valign="middle" bgcolor="#CCCCCC">
          <span class="STYLE10 STYLE18">类别  </span></td>
          <td height="30" align="center" valign="middle" bgcolor="#CCCCCC">
          <span class="STYLE10 STYLE18">价格  </span></td>
          <td height="30" align="center" valign="middle" bgcolor="#CCCCCC">
          <span class="STYLE10 STYLE18">库存  </span></td>
          <td height="30" align="center" valign="middle" bgcolor="#CCCCCC">
```

```
        <span class="STYLE18"></span></td>
      </tr>
      <%
        ResultSet rs=null;
        db.connection();
        String query="select * from book";
        rs=db.query(query);
        while (rs.next())
          { String s1=rs.getString(1);
            String s2=rs.getString(2);
            String s3=rs.getString(3);
            String s4=rs.getString(4);
            String s5=rs.getString(5);
            String s6=rs.getString(6);
            String s7=rs.getString(7);
      %>
      <tr>
        <td height="30" align="center" valign="middle"><span class="STYLE18">
        <%=s1%></span></td>
        <td height="30" align="center" valign="middle"><span class="STYLE18">
        <%=s2%></span></td>
        <td height="30" align="center" valign="middle"><span class="STYLE18">
        <%=s3%></span></td>
        <td height="30" align="center" valign="middle"><span class="STYLE18">
        <%=s4%></span></td>
        <td height="30" align="center" valign="middle"><span class="STYLE18">
        <%=s5%></span></td>
        <td height="30" align="center" valign="middle"><span class="STYLE18">
        <%=s6%></span></td>
        <td height="30" align="center" valign="middle"><span class="STYLE18">
        <%=s7%></span></td>
        <td height="30" align="center" valign="middle"><span class="STYLE10
        STYLE18"><span class="STYLE14"><a href="bookdelete1.jsp?id=<%=
        s1%>">删除</a></span> </span></td>
      </tr><% } %>
    </table>
  </form>
  <%
    String id=request.getParameter("id");
    if (id!=null)
   {
     String sql="DELETE FROM book WHERE id="+"\'"+id+"\'";
     db.connection();
     db.delete(sql);
     response.sendRedirect("bookdelete.jsp");
   }
  %>
 <!-- InstanceEndEditable --></td>
</tr>
```

```
<tr>
 <td height="30" align="center" bgcolor="#FFCC66"><span class="STYLE12"><a
 href="bookadd.jsp">添加图书</a></span></td>
</tr>
<tr>
 <td height="30" align="center" bgcolor="#FFCC66"><span class="STYLE12"><a
 href="bookedit.jsp">修改图书</a></span></td>
</tr>
<tr>
 <td height="30" align="center" bgcolor="#FFCC66"><span class="STYLE12"><a
 href="bookdelete.jsp">删除图书</a></span></td>
</tr>
<tr>
 <td height="30" align="center" bgcolor="#FFCC66"><span class="STYLE12"><a
 href="bookquery.jsp">查询图书</a></span></td>
</tr>
<tr>
 <td height="40" class="STYLE2">订单管理</td>
</tr>
<tr>
 <td height="40" class="STYLE2">会员管理</td>
</tr>
<tr>
 <td height="30" align="center" bgcolor="#FFCC66" class="STYLE12">查询会员</td>
</tr>
<tr>
 <td height="30" align="center" bgcolor="#FFCC66" class="STYLE12">删除会员</td>
</tr>
<%
  GregorianCalendar d=new GregorianCalendar();
  String year=Integer.toString(d.get(Calendar.YEAR));
  String month=Integer.toString(d.get(Calendar.MONTH)+1);
  String day=Integer.toString(d.get(Calendar.DAY_OF_MONTH));
%>
<tr>
 <td height="60" align="center"><span class="STYLE10">操作员:<%=(String)
 session.getAttribute("username")%><br />
   时间:<%=year%>-<%=month%>-<%=day%></span></td>
</tr>
<tr>
 <td height="30" align="center"><a href="manage.jsp">返回管理员首页</a></td>
</tr>
</table>
</body>
<!-- InstanceEnd --></html>
```

【bookdelete1.jsp】

```
<%@ page contentType="text/html; charset=gb2312" language="java" import="java.sql.*,
java.util.*" errorPage="" %>
```

```
<html>
  <head></head>
  <body>
   <jsp:useBean id="db" class="bean.mydb" scope="page"/>
   <%
    String id=request.getParameter("id");
    if (id!=null)
    {
      String sql="DELETE FROM book WHERE id="+"\'"+id+"\'";
      db.connection();
      db.delete(sql);
    response.sendRedirect("bookdelete.jsp");
    } %>
  </body>
</html>
```

【bookedit.jsp】

```
<%@ page contentType="text/html; charset=gb2312" language="java" import="java.sql.*,
java.util.*" errorPage="" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.
w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"><!-- InstanceBegin template=
"/Templates/mb.dwt.jsp" codeOutsideHTMLIsLocked="false" -->
 <head>
  <meta http-equiv="Content-Type" content="text/html; charset=gb2312" />
  <!-- InstanceBeginEditable name="doctitle" -->
  <title>无标题文档</title>
  <!-- InstanceEndEditable -->
  <!-- InstanceBeginEditable name="head" -->
  <!-- InstanceEndEditable -->
  <style type="text/css">
   <!--
     .STYLE14 {color: #FF6600}
     .STYLE18 {font-size: 24}
   -->
  </style>
  <style type="text/css">
   <!--
     .STYLE10 {font-family: "宋体"}
     .STYLE12 {font-family: Arial, Helvetica, sans-serif; font-size: 16px; }
     .STYLE2 {font-family: "华文行楷";
       font-size: 24px;
       color: #FF9900;
       }
   -->
  </style>
</head>
<jsp:useBean id="db" class="bean.mydb" scope="page"/>
<body>
```

```
<table width="900" border="0" align="center" cellpadding="0" cellspacing="0">
 <tr>
  <td width="149" height="40"><span class="STYLE2">图书管理</span></td>
   < td width = " 751 " rowspan = " 10 " align = " center " valign = " top " > <!- -
   InstanceBeginEditable name="EditRegion1" -->
  <form id="form1" name="form1" method="post" action="">
    <table width="700" border="0" cellspacing="0" cellpadding="0">
      <tr>
        <td height="30" align="center" valign="middle" bgcolor="#CCCCCC"><span
        class="STYLE10 STYLE18"> 图书 ID</span></td>
        <td height="30" align="center" valign="middle" bgcolor="#CCCCCC"><span
        class="STYLE10 STYLE18">图书名称  </span></td>
        <td height="30" align="center" valign="middle" bgcolor="#CCCCCC"><span
        class="STYLE10 STYLE18">作者  </span></td>
        <td height="30" align="center" valign="middle" bgcolor="#CCCCCC"><span
        class="STYLE10 STYLE18">出版社  </span></td>
        <td height="30" align="center" valign="middle" bgcolor="#CCCCCC"><span
        class="STYLE10 STYLE18">类别  </span></td>
        <td height="30" align="center" valign="middle" bgcolor="#CCCCCC"><span
        class="STYLE10 STYLE18">价格  </span></td>
        <td height="30" align="center" valign="middle" bgcolor="#CCCCCC"><span
        class="STYLE10 STYLE18">库存  </span></td>
        <td height="30" align="center" valign="middle" bgcolor="#CCCCCC"><span
        class="STYLE18"></span></td>
      </tr>
      <%
        ResultSet rs=null;
        db.connection();
        String query="select * from book";
        rs=db.query(query);
        while (rs.next())
        {String s1=rs.getString(1);
            String s2=rs.getString(2);
            String s3=rs.getString(3);
            String s4=rs.getString(4);
            String s5=rs.getString(5);
            String s6=rs.getString(6);
            String s7=rs.getString(7);
      %>
      <tr>
        <td height="30" align="center" valign="middle"> <%=s1%></td>
        <td height="30" align="center" valign="middle"> <%=s2%></td>
        <td height="30" align="center" valign="middle"> <%=s3%></td>
        <td height="30" align="center" valign="middle"> <%=s4%></td>
        <td height="30" align="center" valign="middle"> <%=s5%></td>
        <td height="30" align="center" valign="middle"> <%=s6%></td>
        <td height="30" align="center" valign="middle"> <%=s7%></td>
        <td height = " 30 " align = " center " valign = " middle " > < span class = " STYLE10
        STYLE18"> < span class = " STYLE14 " > < a href = " bookedit1. jsp? id = <% =
```

```
            s1%>" >修改</a></span> </span></td>
          </tr>
          <% } %>
        </table>
      </form>
    <!-- InstanceEndEditable --></td>
  </tr>
  <tr>
    <td height="30" align="center" bgcolor="#FFCC66"><span class="STYLE12"><a
    href="bookadd.jsp">添加图书</a></span></td>
  </tr>
  <tr>
    <td height="30" align="center" bgcolor="#FFCC66"><span class="STYLE12"><a
    href="bookedit.jsp">修改图书</a></span></td>
  </tr>
  <tr>
    <td height="30" align="center" bgcolor="#FFCC66"><span class="STYLE12"><a
    href="bookdelete.jsp">删除图书</a></span></td>
  </tr>
  <tr>
    <td height="30" align="center" bgcolor="#FFCC66"><span class="STYLE12"><a
    href="bookquery.jsp">查询图书</a></span></td>
  </tr>
  <tr>
    <td height="40" class="STYLE2">订单管理</td>
  </tr>
  <tr>
    <td height="40" class="STYLE2">会员管理</td>
  </tr>
  <tr>
    <td height="30" align="center" bgcolor="#FFCC66" class="STYLE12">查询会员</td>
  </tr>
  <tr>
    <td height="30" align="center" bgcolor="#FFCC66" class="STYLE12">删除会员</td>
  </tr>
  <%
    GregorianCalendar d=new GregorianCalendar();
    String year=Integer.toString(d.get(Calendar.YEAR));
    String month=Integer.toString(d.get(Calendar.MONTH)+1);
    String day=Integer.toString(d.get(Calendar.DAY_OF_MONTH));
  %>

  <tr>
    <td height="60" align="center"><span class="STYLE10">操作员:<%=(String)
    session.getAttribute("username")%><br />
      时间:<%=year%>-<%=month%>-<%=day%></span></td>
  </tr>
  <tr>
    <td height="30" align="center"><a href="manage.jsp">返回管理员首页</a></td>
```

```
    </tr>
  </table>

 </body>
<!-- InstanceEnd --></html>
```

【bookedit1.jsp】

```
<%@ page contentType="text/html; charset=gb2312" language="java" import="java.sql.*,
java.util.*" errorPage="" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.
w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"><!-- InstanceBegin template=
"/Templates/mb.dwt.jsp" codeOutsideHTMLIsLocked="false" -->
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=gb2312" />
  <!-- InstanceBeginEditable name="doctitle" -->
  <title>无标题文档</title>
  <!-- InstanceEndEditable -->
  <!-- InstanceBeginEditable name="head" -->
  <style type="text/css">
   <!--
    .STYLE19 {font-family: "宋体"; font-size: 16; }
    .STYLE19 {font-family: Arial, Helvetica, sans-serif; font-size: 16px; }
   -->
  </style>
  <!-- InstanceEndEditable -->
  <style type="text/css">
   <!--
    .STYLE14 {color: #FF6600}
    .STYLE18 {font-size: 24}
   -->
  </style>
  <style type="text/css">
   <!--
    .STYLE10 {font-family: "宋体"}
    .STYLE12 {font-family: Arial, Helvetica, sans-serif; font-size: 16px; }
    .STYLE2 {font-family: "华文行楷";
      font-size: 24px;
      color: #FF9900;
      }
   -->
  </style>
</head>
<jsp:useBean id="db" class="bean.mydb" scope="page"/>
<body>
  <table width="900" border="0" align="center" cellpadding="0" cellspacing="0">
   <tr>
    <td width="149" height="40"><span class="STYLE2">图书管理</span></td>
     <td width="751" rowspan="10" align="center" valign="top"><!--
```

```
InstanceBeginEditable name="EditRegion1" -->
<form id="form1" name="form1" method="post" action="">
 <table width="500" border="0" align="center" cellpadding="0" cellspacing="0">
   <tr>
       <td height="30" colspan="2" align="left" valign="middle" bgcolor=
       "#CCCCCC" class="STYLE7"><label><span class="STYLE10">图书管
       理》》修改图书</span></label></td>
   </tr>
   <%
       String id=request.getParameter("id");
       ResultSet rs=null;
       if (id!=null)
       {//变量sql存放查询语句,按照id查询数据库表
           String sql="SELECT * FROM book WHERE id='"+id+"'";
           //对象db调用connection方法,连接数据库
           db.connection();
           //对象db调用query方法,参数为"按照id查询数据库表"
           rs=db.query(sql);
       while (rs.next())   {
             //提取要修改的详细信息
             String s1=rs.getString(1);
             String s2=rs.getString(2);
             String s3=rs.getString(3);
             String s4=rs.getString(4);
             String s6=rs.getString(6);
             String s7=rs.getString(7);
       %>
   <tr>
     <td height="15" align="right" valign="middle" class="STYLE7 STYLE8">图书
     ID:</td>
     <td height="30" align="left" valign="middle"><input name="bookid" type=
     "text" id="textfield" value="<%=s1%>"/></td>
   </tr>
   <tr>
     <td height="30" align="right" valign="middle" class="STYLE7 STYLE8">图书
     名称:</td>
     <td height="30" align="left" valign="middle" ><span class="STYLE19">
       <label>
        <input type="text" name="bookname" id="textfield2" value="<%=s2%>"/>
       </label>
     </span></td>
   </tr>
   <tr>
     <td height="30" align="right" valign="middle" class="STYLE7 STYLE8">作
     者:</td>
     <td height="30" align="left" valign="middle"><span class="STYLE19">
       <label>
        <input type="text" name="author" id="textfield3" value="<%=s3%>"/>
       </label>
```

```
  </span></td>
</tr>
<tr>
  <td height="30" align="right" valign="middle" class="STYLE7 STYLE8">出版
  社:</td>
  <td height="30" align="left" valign="middle"><span class="STYLE19">
    <label>
     <input type="text" name="publisher" id="textfield4" value="<%=s4%>"/>
    </label>
  </span></td>
</tr>
<tr>
  <td height="30" align="right" valign="middle" class="STYLE7 STYLE8">类
  别:</td>
  <td height="30" align="left" valign="middle"><span class="STYLE19">
    <label>
     <select name="category" id="select">
       <option selected="selected">科技</option>
       <option>艺术</option>
       <option>生活</option>
       <option>社会</option>
       <option>娱乐</option>
       <option>教育</option>
     </select>
    </label>
  </span></td>
</tr>
<tr>
  <td height="30" align="right" valign="middle" class="STYLE7 STYLE8">价
  格:</td>
  <td height="30" align="left" valign="middle"><span class="STYLE19">
    <label>
     <input type="text" name="price" id="textfield5" value="<%=s6%>"/>
    </label>
  </span></td>
</tr>
<tr>
  <td height="30" align="right" valign="middle" class="STYLE7 STYLE8">库
  存:</td>
  <td height="30" align="left" valign="middle"><span class="STYLE19">
    <label>
     <input type="text" name="stock" id="textfield6" value="<%=s7%>"/>
    </label>
  </span></td>
 </tr>
<% } }%>
<tr>
  <td height="30" colspan="2" align="center" valign="middle"><label>
    <input type="submit" name="button" id="button" value="确定" />
```

```
                </label>
                <label>
                    <input type="reset" name="button2" id="button2" value="重填" />
                </label></td>
            </tr>
          </table>
        </form>
      <!-- InstanceEndEditable --></td>
    </tr>
    <tr>
      <td height="30" align="center" bgcolor="#FFCC66"><span class="STYLE12"><a
      href="bookadd.jsp">添加图书</a></span></td>
    </tr>
    <tr>
      <td height="30" align="center" bgcolor="#FFCC66"><span class="STYLE12"><a
      href="bookedit.jsp">修改图书</a></span></td>
    </tr>
    <tr>
      <td height="30" align="center" bgcolor="#FFCC66"><span class="STYLE12"><a
      href="bookdelete.jsp">删除图书</a></span></td>
    </tr>
    <tr>
      <td height="30" align="center" bgcolor="#FFCC66"><span class="STYLE12"><a
      href="bookquery.jsp">查询图书</a></span></td>
    </tr>
    <tr>
      <td height="40" class="STYLE2">订单管理</td>
    </tr>
    <tr>
      <td height="40" class="STYLE2">会员管理</td>
    </tr>
    <tr>
      <td height="30" align="center" bgcolor="#FFCC66" class="STYLE12">查询会员</td>
    </tr>
    <tr>
      <td height="30" align="center" bgcolor="#FFCC66" class="STYLE12">删除会员</td>
    </tr>
    <%
      GregorianCalendar d=new GregorianCalendar();
      String year=Integer.toString(d.get(Calendar.YEAR));
      String month=Integer.toString(d.get(Calendar.MONTH)+1);
      String day=Integer.toString(d.get(Calendar.DAY_OF_MONTH));
    %>

    <tr>
      <td height="60" align="center"><span class="STYLE10">操作员:<%=(String)
      session.getAttribute("username")%><br />
        时间:<%=year%>-<%=month%>-<%=day%></span></td>
    </tr>
```

```
    <tr>
     <td height="30" align="center"><a href="manage.jsp">返回管理员首页</a></td>
    </tr>
   </table>
   <%
    //获取表单提交数据
    String bookid=db.hzsb(request.getParameter("bookid")) ;
    String bookname=db.hzsb(request.getParameter("bookname")) ;
    String author=db.hzsb(request.getParameter("author"));
    String publisher=db.hzsb(request.getParameter("publisher"));
    String category=db.hzsb(request.getParameter("category"));
    String price=db.hzsb(request.getParameter("price"));
    String stock=db.hzsb(request.getParameter("stock"));

    String rssql = "UPDATE book SET name = '" + bookname + "', author = '" + author +
    "',publisher='"+publisher+"',type='"+category+"',price='"+price+"',stock='"+
    stock+"' WHERE id='"+bookid+"'";
    db.connection();
    //out.println(rssql);
    int rsn=db.update(rssql);
    if (rsn>0)
      response.sendRedirect("bookedit.jsp");
   %>
  </body>
<!-- InstanceEnd --></html>
```

【bookquery.jsp】

```
<%@ page contentType="text/html; charset=gb2312" language="java" import="java.sql.*,
java.util.*" errorPage="" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.
w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"><!-- InstanceBegin template=
"/Templates/mb.dwt.jsp" codeOutsideHTMLIsLocked="false" -->
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=gb2312" />
  <!-- InstanceBeginEditable name="doctitle" -->
  <title>无标题文档</title>
  <!-- InstanceEndEditable -->
  <!-- InstanceBeginEditable name="head" -->
  <style type="text/css">
   <!--
    .STYLE11 {font-size: 18px}
   -->
  </style>
  <!-- InstanceEndEditable -->
  <style type="text/css">
   <!--
    .STYLE14 {color: #FF6600}
    .STYLE18 {font-size: 24}
```

```
  -->
  </style>
  <style type="text/css">
   <!--
    .STYLE10 {font-family: "宋体"}
    .STYLE12 {font-family: Arial, Helvetica, sans-serif; font-size: 16px; }
    .STYLE2 {font-family: "华文行楷";
      font-size: 24px;
      color: #FF9900;
      }
   -->
  </style>
</head>
<jsp:useBean id="db" class="bean.mydb" scope="page"/>
<body>
  <table width="900" border="0" align="center" cellpadding="0" cellspacing="0">
   <tr>
    <td width="149" height="40"><span class="STYLE2">图书管理</span></td>
     < td width = " 751 " rowspan = " 10 " align = " center " valign = " top " > <!- -
     InstanceBeginEditable name="EditRegion1" -->
     <form id="form1" name="form1" method="post" action="bookquery1.jsp">
      <table width="500" border="0" align="center" cellpadding="0" cellspacing="0">
        <tr>
            < td height = " 30 " colspan = " 2 " align = " left " valign = " middle " bgcolor =
            "#CCCCCC" class="STYLE7"><span class="STYLE10 STYLE11">
            <label>图书管理»»查询图书</label>
          </span></td>
        </tr>
        <tr>
          <td height = " 15 " align = " right " valign = " middle " class = " STYLE7 STYLE8
          STYLE11 STYLE11 STYLE8">图书 ID:</td>
          <td height="30" align="left" valign="middle"><input name="bookid" type=
          "text" class="STYLE11" id="textfield" /></td>
        </tr>
        <tr>
          <td height = " 30 " align = " right " valign = " middle " class = " STYLE7 STYLE8
          STYLE11 STYLE11 STYLE8">图书名称:</td>
          <td height="30" align="left" valign="middle"><span class="STYLE10 STYLE11">
            <label>
             <input type="text" name="bookname" id="textfield2" />
            </label>
          </span></td>
        </tr>
        <tr>
          <td height = " 30 " align = " right " valign = " middle " class = " STYLE7 STYLE8
          STYLE11 STYLE11 STYLE8">作者:</td>
          <td height="30" align="left" valign="middle"><span class="STYLE10 STYLE11">
            <label>
             <input type="text" name="author" id="textfield3" />
```

```
          </label>
        </span></td>
      </tr>
      <tr>
        <td height = " 30 "  align = " right "  valign = " middle "  class = " STYLE7  STYLE8
        STYLE11 STYLE11 STYLE8">出版社:</td>
        <td height="30" align="left" valign="middle"><span class="STYLE10 STYLE11">
          <label>
           <input type="text" name="publisher" id="textfield4" />
          </label>
        </span></td>
      </tr>
      <tr>
        <td height = " 30 "  align = " right "  valign = " middle "  class = " STYLE7  STYLE8
        STYLE11 STYLE11 STYLE8">类别:</td>
        <td height="30" align="left" valign="middle"><span class="STYLE10 STYLE11">
          <label>
           <select name="category" id="select">
            <option></option>
            <option>科技</option>
            <option>艺术</option>
            <option>生活</option>
            <option>社会</option>
            <option>娱乐</option>
            <option>教育</option>
           </select>
          </label>
        </span></td>
      </tr>
      <tr>
        <td height="30" colspan="2" align="center" valign="middle"><span class=
        "STYLE10">
          <label>
            <input type="submit" name="button" id="button" value="确定" />
          </label>
          <label></label>
        </span></td>
      </tr>
    </table>
  </form>
  <!-- InstanceEndEditable --></td>
</tr>
<tr>
  <td height="30" align="center" bgcolor="#FFCC66"><span class="STYLE12"><a
  href="bookadd.jsp">添加图书</a></span></td>
</tr>
<tr>
  <td height="30" align="center" bgcolor="#FFCC66"><span class="STYLE12"><a
  href="bookedit.jsp">修改图书</a></span></td>
```

```
      </tr>
      <tr>
       <td height="30" align="center" bgcolor="#FFCC66"><span class="STYLE12"><a
       href="bookdelete.jsp">删除图书</a></span></td>
      </tr>
      <tr>
       <td height="30" align="center" bgcolor="#FFCC66"><span class="STYLE12"><a
       href="bookquery.jsp">查询图书</a></span></td>
      </tr>
      <tr>
       <td height="40" class="STYLE2">订单管理</td>
      </tr>
      <tr>
       <td height="40" class="STYLE2">会员管理</td>
      </tr>
      <tr>
       <td height="30" align="center" bgcolor="#FFCC66" class="STYLE12">查询会员</td>
      </tr>
      <tr>
       <td height="30" align="center" bgcolor="#FFCC66" class="STYLE12">删除会员</td>
      </tr>
      <%
        GregorianCalendar d=new GregorianCalendar();
        String year=Integer.toString(d.get(Calendar.YEAR));
        String month=Integer.toString(d.get(Calendar.MONTH)+1);
        String day=Integer.toString(d.get(Calendar.DAY_OF_MONTH));
      %>

      <tr>
       <td height="60" align="center"><span class="STYLE10">操作员:<%=(String)
       session.getAttribute("username")%><br />
         时间:<%=year%>-<%=month%>-<%=day%></span></td>
      </tr>
      <tr>
       <td height="30" align="center"><a href="manage.jsp">返回管理员首页</a></td>
      </tr>
    </table>

</body>
<!-- InstanceEnd --></html>
```

【bookquery1.jsp】

```
<%@ page contentType="text/html; charset=gb2312" language="java" import="java.sql.*,
java.util.*" errorPage="" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.
w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"><!-- InstanceBegin template=
"/Templates/mb.dwt.jsp" codeOutsideHTMLIsLocked="false" -->
 <head>
```

```
<meta http-equiv="Content-Type" content="text/html; charset=gb2312" />
<!-- InstanceBeginEditable name="doctitle" -->
<title>查询图书</title>
<!-- InstanceEndEditable -->
<!-- InstanceBeginEditable name="head" -->
<style type="text/css">
 <!--
  .STYLE14 {color: #FF6600}
  .STYLE18 {font-size: 24}
 -->
</style>
<!-- InstanceEndEditable -->
<style type="text/css">
 <!--
  .STYLE10 {font-family: "宋体"}
  .STYLE12 {font-family: Arial, Helvetica, sans-serif; font-size: 16px; }
  .STYLE2 {font-family: "华文行楷";
     font-size: 24px;
     color: #FF9900;
     }
 -->
</style>
</head>
<jsp:useBean id="db" class="bean.mydb" scope="page"/>
<body>
 <table width="900" border="0" align="center" cellpadding="0" cellspacing="0">
  <tr>
   <td width="149" height="40"><span class="STYLE2">图书管理</span></td>
    < td width = " 751 " rowspan = " 10 " align = " center " valign = " top " > <!- -
    InstanceBeginEditable name="EditRegion3" -->
     <form id="form1" name="form1" method="post" action="">
       <table width="700" border="0" cellspacing="0" cellpadding="0">
         <tr>
           <td height="30" align="center" valign="middle" bgcolor="#CCCCCC">
           <span class="STYLE10 STYLE18"> 图书 ID</span></td>
           <td height="30" align="center" valign="middle" bgcolor="#CCCCCC">
           <span class="STYLE10 STYLE18">图书名称  </span></td>
           <td height="30" align="center" valign="middle" bgcolor="#CCCCCC">
           <span class="STYLE10 STYLE18">作者  </span></td>
           <td height="30" align="center" valign="middle" bgcolor="#CCCCCC">
           <span class="STYLE10 STYLE18">出版社  </span></td>
           <td height="30" align="center" valign="middle" bgcolor="#CCCCCC">
           <span class="STYLE10 STYLE18">类别  </span></td>
           <td height="30" align="center" valign="middle" bgcolor="#CCCCCC">
           <span class="STYLE10 STYLE18">价格  </span></td>
           <td height="30" align="center" valign="middle" bgcolor="#CCCCCC">
           <span class="STYLE10 STYLE18">库存  </span></td>
           <td height="30" align="center" valign="middle" bgcolor="#CCCCCC">
           <span class="STYLE18"></span></td>
```

```
            </tr>
            <%
              String id=db.hzsb(request.getParameter("bookid")) ;
              String bookname=db.hzsb(request.getParameter("bookname"));
              String author=db.hzsb(request.getParameter("author"));
              String publisher=db.hzsb(request.getParameter("publisher"));
              String category=db.hzsb(request.getParameter("category"));
              ResultSet rs=null;
              String sqls="SELECT * FROM book WHERE id='"+id+"' OR name='"+
              bookname+"' OR author='"+author+"' OR publisher='"+publisher+"' OR type='
              "+category+"'";
              db.connection();
              rs=db.query(sqls);
              //out.println(sqls);
              while (rs.next())
                { String s1=rs.getString(1);
                  String s2=rs.getString(2);
                  String s3=rs.getString(3);
                  String s4=rs.getString(4);
                  String s5=rs.getString(5);
                  String s6=rs.getString(6);
                  String s7=rs.getString(7);
              %>
              <tr>
                <td height="30" align="center" valign="middle"><span class="STYLE18">
                <%=s1%></span></td>
                <td height="30" align="center" valign="middle"><span class="STYLE18">
                <%=s2%></span></td>
                <td height="30" align="center" valign="middle"><span class="STYLE18">
                <%=s3%></span></td>
                <td height="30" align="center" valign="middle"><span class="STYLE18">
                <%=s4%></span></td>
                <td height="30" align="center" valign="middle"><span class="STYLE18">
                <%=s5%></span></td>
                <td height="30" align="center" valign="middle"><span class="STYLE18">
                <%=s6%></span></td>
                <td height="30" align="center" valign="middle"><span class="STYLE18">
                <%=s7%></span></td>

              </tr><% } %>
          </table>
        </form>

      <!-- InstanceEndEditable --></td>
    </tr>
    <tr>
     <td height="30" align="center" bgcolor="#FFCC66"><span class="STYLE12"><a
     href="bookadd.jsp">添加图书</a></span></td>
    </tr>
```

```
<tr>
  <td height="30" align="center" bgcolor="#FFCC66"><span class="STYLE12"><a
  href="bookedit.jsp">修改图书</a></span></td>
</tr>
<tr>
  <td height="30" align="center" bgcolor="#FFCC66"><span class="STYLE12"><a
  href="bookdelete.jsp">删除图书</a></span></td>
</tr>
<tr>
  <td height="30" align="center" bgcolor="#FFCC66"><span class="STYLE12"><a
  href="bookquery.jsp">查询图书</a></span></td>
</tr>
<tr>
  <td height="40" class="STYLE2">订单管理</td>
</tr>
<tr>
  <td height="40" class="STYLE2">会员管理</td>
</tr>
<tr>
  <td height="30" align="center" bgcolor="#FFCC66" class="STYLE12">查询会员</td>
</tr>
<tr>
  <td height="30" align="center" bgcolor="#FFCC66" class="STYLE12">删除会员</td>
</tr>
<%
   GregorianCalendar d=new GregorianCalendar();
   String year=Integer.toString(d.get(Calendar.YEAR));
   String month=Integer.toString(d.get(Calendar.MONTH)+1);
   String day=Integer.toString(d.get(Calendar.DAY_OF_MONTH));
%>
<tr>
  <td height="60" align="center"><span class="STYLE10">操作员:<%=(String)
  session.getAttribute("username")%><br />
    时间:<%=year%>-<%=month%>-<%=day%></span></td>
</tr>
<tr>
  <td height="30" align="center"><a href="manage.jsp">返回管理员首页</a></td>
</tr>
</table>
</body>
<!-- InstanceEnd --></html>
```

【userregister.jsp】

```
<%@ page contentType="text/html; charset=gb2312" language="java" import="java.sql.*"
errorPage="" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.
w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
 <head>
```

```
  <meta http-equiv="Content-Type" content="text/html; charset=gb2312" />
  <title>用户注册</title>
  <style type="text/css">
   <!--
    .STYLE1 {
      font-family: "华文新魏";
      font-size: 24px;
      color: #FF9900;
    }
    .STYLE4 {font-family: "宋体"; font-size: 18px; }
   -->
  </style>
</head>
<jsp:useBean id="db" class="bean.mydb" scope="page"/>
<body>
  <form id="form1" name="form1" method="post" action="">
   <table width="400" border="0" align="center" cellpadding="0" cellspacing="2" bgcolor
   ="#CCCCCC">
     <tr>
      <td height="40" colspan="2" align="center" valign="middle" bgcolor="#FFFFFF">
      <span class="STYLE1">用户注册</span></td>
     </tr>
     <tr>
      <td width="40%" height="30" align="center" bgcolor="#FFFFFF"><span class=
      "STYLE4">用户名:</span></td>
      <td height="30" align="left" bgcolor="#FFFFFF"><label>
        <input type="text" name="username" id="textfield" />
       </label></td>
     </tr>
     <tr>
      <td height="30" align="center" bgcolor="#FFFFFF"><span class="STYLE4">密
      码:</span></td>
      <td height="30" align="left" bgcolor="#FFFFFF"><label>
        <input type="text" name="password" id="textfield2" />
       </label></td>
     </tr>
     <tr>
      <td height="30" align="center" bgcolor="#FFFFFF"><span class="STYLE4">确
      认密码:</span></td>
      <td height="30" align="left" bgcolor="#FFFFFF"><label>
        <input type="text" name="confirmpassword" id="textfield3" />
       </label></td>
     </tr>
     <tr>
      <td height="30" align="center" bgcolor="#FFFFFF"><span class="STYLE4">姓
      名:</span></td>
      <td height="30" align="left" bgcolor="#FFFFFF"><label>
        <input type="text" name="name" id="textfield4" />
       </label></td>
```

```
</tr>
<tr>
 <td height="30" align="center" bgcolor="#FFFFFF"><span class="STYLE4">性
 别:</span></td>
 <td height="30" align="left" bgcolor="#FFFFFF"><label>
   <select name="sex" id="select">
    <option>男</option>
    <option>女</option>
   </select>
  </label></td>
</tr>
<tr>
 <td height="30" align="center" bgcolor="#FFFFFF"><span class="STYLE4">地
 址:</span></td>
 <td height="30" align="left" bgcolor="#FFFFFF"><label>
   <input type="text" name="address" id="textfield6" />
  </label></td>
</tr>
<tr>
 <td height="30" align="center" bgcolor="#FFFFFF"><span class="STYLE4">邮
 编:</span></td>
 <td height="30" align="left" bgcolor="#FFFFFF"><label>
   <input type="text" name="code" id="textfield7" />
  </label></td>
</tr>
<tr>
 <td height="30" align="center" bgcolor="#FFFFFF"><span class="STYLE4">电
 话:</span></td>
 <td height="30" align="left" bgcolor="#FFFFFF"><label>
   <input type="text" name="tel" id="textfield8" />
  </label></td>
</tr>
<tr>
 <td height="30" align="center" bgcolor="#FFFFFF"><span class="STYLE4">
 e-mail:</span></td>
 <td height="30" align="left" bgcolor="#FFFFFF"><label>
   <input type="text" name="email" id="textfield9" />
  </label></td>
</tr>
<tr>
 <td height="30" colspan="2" align="center" valign="middle" bgcolor="#FFFFFF">
 <label>
    <input type="submit" name="button" id="button" value="确定" /> 
       
  </label>
  <label>
   <input type="reset" name="button2" id="button2" value="重填" />
  </label></td>
</tr>
```

```
  </table>
 </form>
 <%
   String id=db.hzsb(request.getParameter("username")) ;
   String password=db.hzsb(request.getParameter("password"));
   String name=db.hzsb(request.getParameter("name"));
   String sex=db.hzsb(request.getParameter("sex"));
   String address=db.hzsb(request.getParameter("address"));
   String code=db.hzsb(request.getParameter("code"));
   String tel=db.hzsb(request.getParameter("tel"));
   String email=db.hzsb(request.getParameter("email"));

   if (id!=null && password!=null && name!=null && sex!=null && address!=null
            && code!=null && tel!=null && email!=null)
     {
       String rssql="insert into customer values("+"\'"+id+"\'"+","+"\'"+password+"\
       '"+","+"\'"+name+"\'"+","+"\'"+sex+"\'"+","+"\'"+address+"\'"+","
       +"\'"+code+"\'"+","+"\'"+tel+"\'"+","+"\'"+email+"\'"+")";
       db.connection();
       int rsn=db.update(rssql);
        if (rsn>0)
          response.sendRedirect("../index.html");
   }
  %>
 </body>
</html>
```

【cartmanage.jsp】

```
<%@ page contentType="text/html; charset=gb2312" language="java" import="java.sql.*,
java.util.*" errorPage="" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.
w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"><!-- InstanceBegin template=
"/Templates/cart.dwt.jsp" codeOutsideHTMLIsLocked="false" -->
 <head>
  <meta http-equiv="Content-Type" content="text/html; charset=gb2312" />
  <!-- InstanceBeginEditable name="doctitle" -->
  <title>购书</title>
  <!-- InstanceEndEditable -->
  <!-- InstanceBeginEditable name="head" -->
  <style type="text/css">
   <!--
    .STYLE14 {color: #FF6600}
    .STYLE18 {font-size: 24}
   -->
  </style>
  <!-- InstanceEndEditable -->
  <style type="text/css">
   <!--
```

```
    .STYLE10 {font-family: "宋体"}
    .STYLE12 {font-family: Arial, Helvetica, sans-serif; font-size: 16px; }
    .STYLE2 {font-family: "华文行楷";
      font-size: 24px;
      color: #FF9900;
      }
    body {
        margin-left: 0px;
        margin-top: 0px;
        margin-right: 0px;
        margin-bottom: 0px;
      }
  -->
  </style>
</head>
<jsp:useBean id="db" class="bean.mydb" scope="page"/>
<body>
  <table width="900" border="0" align="center" cellpadding="0" cellspacing="0">
   <tr>
    <td width="149" height="130"><span class="STYLE2"><%=session.getAttribute("
    username")%>欢迎光临本网站选购图书</span></td>
    <td width="751" rowspan="7" align="center" valign="top"><!-- InstanceBeginEditable
    name="EditRegion3" --> 

       <table width="700" border="0" cellspacing="0" cellpadding="0">
          <tr>
            <td height="30" align="center" valign="middle" bgcolor="#CCCCCC">
            <span class="STYLE10 STYLE18"> 图书 ID</span></td>
            <td height="30" align="center" valign="middle" bgcolor="#CCCCCC">
            <span class="STYLE10 STYLE18">图书名称  </span></td>
            <td height="30" align="center" valign="middle" bgcolor="#CCCCCC">
            <span class="STYLE10 STYLE18">作者  </span></td>
            <td height="30" align="center" valign="middle" bgcolor="#CCCCCC">
            <span class="STYLE10 STYLE18">出版社  </span></td>
            <td height="30" align="center" valign="middle" bgcolor="#CCCCCC">
            <span class="STYLE10 STYLE18"> 库存</span></td>
            <td height="30" align="center" valign="middle" bgcolor="#CCCCCC">
            <span class="STYLE10 STYLE18"> 价格</span></td>
            <td height="30" align="center" valign="middle" bgcolor="#CCCCCC">
            <span class="STYLE10 STYLE18">订购数量  </span><span class=
            "STYLE18"></span></td>

          </tr>
          <%
            ResultSet rs=null;
            db.connection();
            String query="select * from book";
            rs=db.query(query);
            while (rs.next())
```

```
{ String s1=rs.getString(1);
  String s2=rs.getString(2);
  String s3=rs.getString(3);
  String s4=rs.getString(4);
  String s6=rs.getString(6);
  String s7=rs.getString(7);


%>
 <tr>
   <td height="30" align="center" valign="middle"><span class="STYLE18">
   <%=s1%></span></td>
   <td height="30" align="center" valign="middle"><span class="STYLE18">
   <%=s2%></span></td>
   <td height="30" align="center" valign="middle"><span class="STYLE18">
   <%=s3%></span></td>
   <td height="30" align="center" valign="middle"><span class="STYLE18">
   <%=s4%></span></td>
   <td height="30" align="center" valign="middle"><span class="STYLE18">
   <%=s7%></span></td>
   <td height="30" align="center" valign="middle"><span class="STYLE18">
   <%=s6%></span></td>


   <td height="30" align="center" valign="middle"><span class="STYLE18">
   </span>
    <form id="form1" name="form1" method="post" action="purchase.jsp?id=
    <%=s1%>">
      <label><input name="amount" type="text" id="textfield" size="5" /></label>
        <input type="submit" value="购买" name="button2" /><span class=
        "STYLE10 STYLE18"> </span>

    </form>
   </td>
 </tr><% } %>
</table>

  <!-- InstanceEndEditable --></td>
</tr>
<tr>
 <td height="50" align="center" bgcolor="#FFCC66"><a href="cartmanage.jsp">
 <span class="STYLE12">购书</span></a></td>
</tr>
<tr>
 <td height="50" align="center" bgcolor="#FFCC66"><a href="mycart.jsp"><span
 class="STYLE12">我的购物车</span></a></td>
</tr>
<tr>
 <td height="50" align="center" bgcolor="#FFCC66"><a href="orderdisplay.jsp">
```

```
    <span class="STYLE12">查看订单</span></a></td>
   </tr>
   <tr>
    <td height="50" align="center" bgcolor="#FFCC66"><span class="STYLE12">修改
    会员信息</span></td>
   </tr>
   <%
     GregorianCalendar d=new GregorianCalendar();
     String year=Integer.toString(d.get(Calendar.YEAR));
     String month=Integer.toString(d.get(Calendar.MONTH)+1);
     String day=Integer.toString(d.get(Calendar.DAY_OF_MONTH));
   %>
   <tr>
    <td height="40" class="STYLE2"><span class="STYLE10"><br />
     </span><span class="STYLE12">时间:<%=year%>-<%=month%>-<%=
     day%></span></td>
   </tr>
   <tr>
    <td height="30" align="center" class="STYLE12"><a href="cartmanage.jsp">返回会
    员首页</a></td>
   </tr>
  </table>
 </body>
<!-- InstanceEnd --></html>
```

【purchase.jsp】

```
<%@ page contentType="text/html; charset=gb2312" language="java" import="java.sql.*"
errorPage="" %>
<%@page import="java.util.*" %>
<%@page import="bean.cart" %>
<html>
 <jsp:useBean id="db" class="bean.mydb" scope="page"/>
 <body>
  <%
    String bookid=request.getParameter("id");
    String num=request.getParameter("amount");
    boolean match = false; //是否购买过该图书
    Vector list=(Vector)session.getAttribute("cartList");
    cart cart=new cart(bookid,Integer.parseInt(num));
    ResultSet rs=null;
    int stock=0;
    db.connection();
    rs=db.query("SELECT stock FROM book WHERE id='"+bookid+"'");
    while(rs.next())
        stock=Integer.parseInt(rs.getString(1));
    if (stock>=Integer.parseInt(num)){
    if (list==null)// 第一次买书
      {
        list=new Vector();
```

```
            list.addElement(cart);
          }
      else //不是第一次买书
          {
            for(int i=0;i<list.size();i++)
              {
                cart icart =(cart)list.elementAt(i);
                if (bookid.equals(icart.getBookId()))
                  {
                    icart.setNum(icart.getNum()+Integer.parseInt(num));
                    list.setElementAt(icart,i);
                    match=true;
                    break;
                  }//if 结束
              } //for 结束

        if(!match)
            list.addElement(cart);
          }//else 结束

        session.setAttribute("cartList",list);
        response.sendRedirect("mycart.jsp");
      }
      else{
        out.println("库存不足!");
        }
    %>
  </body>
</html>
```

【mycart.jsp】

```
<%@ page contentType="text/html; charset=gb2312" language="java" import="java.sql.*,
java.util.*" errorPage="" %>
<%@ page import="bean.cart"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.
w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"><!-- InstanceBegin template=
"/Templates/cart.dwt.jsp" codeOutsideHTMLIsLocked="false" -->
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=gb2312" />
  <!-- InstanceBeginEditable name="doctitle" -->
  <title>无标题文档</title>
  <!-- InstanceEndEditable -->
  <!-- InstanceBeginEditable name="head" -->
  <style type="text/css">
   <!--
    .STYLE19 {font-size: 18}
   -->
  </style>
```

```
<!-- InstanceEndEditable -->
<style type="text/css">
 <!--
  .STYLE14 {color: #FF6600}
  .STYLE18 {font-size: 24}
 -->
</style>
<style type="text/css">
 <!--
  .STYLE10 {font-family: "宋体"}
  .STYLE12 {font-family: Arial, Helvetica, sans-serif; font-size: 16px; }
  .STYLE2 {font-family: "华文行楷";
    font-size: 24px;
    color: #FF9900;
    }
  body {
      margin-left: 0px;
      margin-top: 0px;
      margin-right: 0px;
      margin-bottom: 0px;
      }
 -->
</style>
</head>
<jsp:useBean id="db" class="bean.mydb" scope="page"/>
<body>
<table width="900" border="0" align="center" cellpadding="0" cellspacing="0">
 <tr>
  <td width="149" height="130"><span class="STYLE2"><%=session.getAttribute
  ("username")%>欢迎光临本网站选购图书</span></td>
  <td width="751" rowspan="7" align="center" valign="top"><!-- InstanceBeginEditable
  name="EditRegion1" -->
    <form id="form1" name="form1" method="post" action="">
     <%
      String bookid=request.getParameter("id");
      String num=request.getParameter("amount");
      if ((Vector)session.getAttribute("cartList")==null){
     %>
      <p>目前你的购物车是空的</p>
     <%
      }else{
      Vector list=(Vector)session.getAttribute("cartList");
      int n=list.size();
     %>
     <table width="700" border="1" cellspacing="0" cellpadding="0">
      <tr>
        <td height="30" align="center" valign="middle"><span class="STYLE10
        STYLE14">图书名称</span></td>
        <td height="30" align="center" valign="middle"><span class="STYLE10
```

```
    STYLE14">作者</span></td>
    <td height="30" align="center" valign="middle"><span class="STYLE10
    STYLE14">出版社</span></td>
    <td height="30" align="center" valign="middle"><span class="STYLE10
    STYLE14">单价(元)</span></td>
    <td height="30" align="center" valign="middle"><span class="STYLE10
    STYLE14">数量</span></td>
   <td height="30" align="center" valign="middle"> </td>
  </tr>
<%
        float money=0;
     for(int i=0;i<n;i++)
       {
       cart c=(cart)list.elementAt(i);
       String sql =" SELECT  id, name, author, publisher, price  "+" FROM  book
       WHERE id ='"+c.getBookId()+"'";
      db.connection();
      ResultSet rs=db.query(sql);
      rs.next();
      //取出记录值
      String s1=rs.getString("id");
      String s2=rs.getString("name");
      String s3=rs.getString("author");
      String s4=rs.getString("publisher");
      String s5=rs.getString("price");
      String price=s5;
      money+=Float.parseFloat(price) * c.getNum();
%>
   <tr>
     <td height="30" align="center"valign="middle"><%=s2%></td>
     <td height="30" align="center"valign="middle"><%=s3%></td>
     <td height="30" align="center"valign="middle"><%=s4%></td>
     <td height="30" align="center"valign="middle"><%=s5%></td>
     <td height="30" align="center" valign="middle"><%=c.getNum()%></td>
     <td height="30" align="center" valign="middle"><a href="cartdelete.jsp?id
     =<%=s1%>">我不要了</a>
     </td>
   </tr>
 <%  }%>
   <tr>
     <td height="30" colspan="4" align="center" valign="middle"><span class=
     "STYLE10 STYLE14">你所选择的图书总额为<%=money%>元</span>
     </td>
     <td height="30" colspan="2" align="center" valign="middle"><span class=
     "STYLE10 STYLE14">

     <ahref="order.jsp?money=<%=Float.toString(money)%>&username=<%=
     (String)session.getAttribute("username")%>">提交订单</a></span>
     </td>
```

```
          </tr>
          <tr>
            <td height="40" colspan="6" align="center" valign="bottom"><span class=
            "STYLE10 STYLE14"><a href="cartclear.jsp">清空购物车</a></span>
            </td>
          </tr>
        </table>
      <%}%>
    </form>
    <!-- InstanceEndEditable --></td>
    </tr>
    <tr>
     <td height="50" align="center" bgcolor="#FFCC66"><a href="cartmanage.jsp">
     <span class="STYLE12">购书</span></a></td>
    </tr>
    <tr>
     <td height="50" align="center" bgcolor="#FFCC66"><a href="mycart.jsp">
     <span class="STYLE12">我的购物车</span></a></td>
    </tr>
    <tr>
     <td height="50" align="center" bgcolor="#FFCC66"><a href="orderdisplay.jsp">
     <span class="STYLE12">查看订单</span></a></td>
    </tr>
    <tr>
     <td height="50" align="center" bgcolor="#FFCC66"><span class="STYLE12">修
     改会员信息</span></td>
    </tr>
    <%
      GregorianCalendar d=new GregorianCalendar();
      String year=Integer.toString(d.get(Calendar.YEAR));
      String month=Integer.toString(d.get(Calendar.MONTH)+1);
      String day=Integer.toString(d.get(Calendar.DAY_OF_MONTH));
    %>
    <tr>
     <td height="40" class="STYLE2"><span class="STYLE10"><br />
      </span><span class="STYLE12">时间:<%=year%>-<%=month%>-<%=
      day%></span></td>
    </tr>
    <tr>
     <td height="30" align="center" class="STYLE12"><a href="cartmanage.jsp">返回
     会员首页</a></td>
    </tr>
  </table>
 </body>
<!-- InstanceEnd --></html>
```

【order.jsp】

```
<%@ page contentType="text/html; charset=gb2312" language="java" import="java.sql.*" %>
<%@page import="java.util.*"%>
```

```
<%@page import="bean.cart"%>
<html>
 <jsp:useBean id="db" class="bean.mydb" scope="page"/>
 <body>
  <%
   int key=0;
   String id="";
   String date="";
   String money=request.getParameter("money");
   String user=request.getParameter("username");
   //获得日期格式为 2011-5-1
   GregorianCalendar d=new GregorianCalendar();
   date=Integer.toString(d.get(Calendar.YEAR))+"-"+Integer.toString(d.get(Calendar.
MONTH)+1)+
        "-"+Integer.toString(d.get(Calendar.DAY_OF_MONTH));
   //查询数据
   String query="SELECT * FROM userorder";
   db.connection();
   ResultSet rs=db.query(query);
   if(!rs.next())
   {
       key=1;    //userorder 中没有数据,也就是没有过订单,则 id 设为 1
   }
   else              //若有数据取出最后 id,再加 1
   {
     rs=db.query(query);
     while(rs.next())
       key=Integer.parseInt(rs.getString("id"));
     key++;
    }
    //获取 id 值,订单号
   id=Integer.toString(key);
   //将订单中的信息存储到 orderlist 表
   Vector list=(Vector)session.getAttribute("cartList");
   int n=list.size();
   out.println(n);
   for (int i=0;i<n;i++)
    {
       cart cart=(cart)list.elementAt(i);
       String sql1="SELECT name,price FROM book WHERE
       id='"+cart.getBookId()+"'";
       rs=db.query(sql1);
       rs.next();
       //获取书名和单价
       String name=rs.getString("name");
       String price=rs.getString("price");
       //将订单中的信息存储到 orderlist 表
       String sql2="INSERT INTO orderlist VALUES("+"\'"+id+"\'"+","+"\'"+user+
       "\'"+","+"\'"+name+"\'"+","+"\'"+cart.getNum()+"\'"+","+"\'"+money+
```

```
            "\'"+","+"\'"+price+"\'"+")";
            //测试语句     out.println(sql2);
            int rsn2=db.update(sql2);
            //if(rsn2>0){out.println("orderlist 已插入");};
            //从 book 表中,减去用户购买的数量
            int stock=0;
            rs=db.query("SELECT stock FROM book WHERE id='"+cart.getBookId()+"'");
              while(rs.next())
              stock=Integer.parseInt(rs.getString(1));
              //购买数量测试语句 out.println(cart.getNum());
              stock=stock-cart.getNum();
              //所剩库存测试语句 out.println(stock);
              String sql3="UPDATE book SET stock="+"\'"+stock+"\'"+" WHERE id="+
              "\'"+cart.getBookId()+"\'";
              //out.println(sql3);
              int rsn3=db.update(sql3);
                }//for 结束
                //插入新的订单到 userorder 表
               String sql="INSERT INTO userorder VALUES("+"\'"+id+"\'"+","+"\'"+user
               +"\'"+","+"\'"+date+"\'"+","+"\'"+money+"\'"+")";
            //测试语句 out.println(sql);
           int rsn1=db.update(sql);
           response.sendRedirect("orderdisplay.jsp");
       %>
     </body>
   </html>
```

【orderdisplay.jsp】

```
<%@ page contentType="text/html; charset=gb2312" language="java" import="java.sql.*,
java.util.*" errorPage="" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.
w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"><!-- InstanceBegin template=
"/Templates/cart.dwt.jsp" codeOutsideHTMLIsLocked="false" -->
 <head>
  <meta http-equiv="Content-Type" content="text/html; charset=gb2312" />
  <!-- InstanceBeginEditable name="doctitle" -->
  <title>无标题文档</title>
  <!-- InstanceEndEditable -->
  <!-- InstanceBeginEditable name="head" -->
  <style type="text/css">
   <!--
    .STYLE19 {font-size: 18}
   -->
  </style>
  <!-- InstanceEndEditable -->
  <style type="text/css">
   <!--
    .STYLE14 {color: #FF6600}
```

```
    .STYLE18 {font-size: 24}
   -->
  </style>
  <style type="text/css">
   <!--
    .STYLE10 {font-family: "宋体"}
    .STYLE12 {font-family: Arial, Helvetica, sans-serif; font-size: 16px; }
    .STYLE2 {font-family: "华文行楷";
      font-size: 24px;
      color: #FF9900;
    }
    body {
        margin-left: 0px;
        margin-top: 0px;
        margin-right: 0px;
        margin-bottom: 0px;
    }
   -->
  </style>
</head>
<jsp:useBean id="db" class="bean.mydb" scope="page"/>
<body>
  <table width="900" border="0" align="center" cellpadding="0" cellspacing="0">
   <tr>
    <td width="149" height="130"><span class="STYLE2"><%=session.getAttribute
    ("username")%>欢迎光临本网站选购图书</span></td>
    <td width="751" rowspan="7" align="center" valign="top"><!-- InstanceBeginEditable
    name="EditRegion1" -->
      <table width="700" border="0" cellspacing="0" cellpadding="0">
        <tr>
          <td height="30" align="center" valign="middle"><span class="STYLE10
          STYLE14">订单号</span></td>
         <td height="30" align="center" valign="middle">用户</td>
         <td height="30" align="center" valign="middle">提交时间</td>
          <td height="30" align="center" valign="middle"><span class="STYLE10
          STYLE14">总金额</span></td>
          <td height="30" align="center" valign="middle"><span class="STYLE10
          STYLE14">详情</span></td>
        </tr>
        <%
              String user=(String)session.getAttribute("username");
                      String query="SELECT * FROM userorder WHERE username=
                      '"+user+"'";
               db.connection();
               ResultSet rs=db.query(query);
               while (rs.next())
               {
                   String s1=rs.getString(1);
                   String s2=rs.getString(2);
```

```
                    String s3=rs.getString(3);
                    String s4=rs.getString(4);
                 %>
            <tr>
              <td height="30" align="center" valign="middle"><span class="STYLE14">
              <%=s1 %></span></td>
              <td height="30" align="center" valign="middle"><span class="STYLE14">
              <%=s2 %></span></td>
              <td height="30" align="center" valign="middle"><span class="STYLE14">
              <%=s3 %></span></td>
              <td height="30" align="center" valign="middle"><span class="STYLE14">
              <%=s4 %></span></td>
              <td height="30" align="center" valign="middle"><span class="STYLE14">
              <a href="orderdetail.jsp?id=<%=s1%>" >查看</a></span></td>
            </tr>
            <%}%>
          </table>
        <!-- InstanceEndEditable --></td>
      </tr>
      <tr>
        <td height="50" align="center" bgcolor="#FFCC66"><a href="cartmanage.jsp">
        <span class="STYLE12">购书</span></a></td>
      </tr>
      <tr>
        <td height="50" align="center" bgcolor="#FFCC66"><a href="mycart.jsp"><span
        class="STYLE12">我的购物车</span></a></td>
      </tr>
      <tr>
        <td height="50" align="center" bgcolor="#FFCC66"><a href="orderdisplay.jsp">
        <span class="STYLE12">查看订单</span></a></td>
      </tr>
      <tr>
        <td height="50" align="center" bgcolor="#FFCC66"><span class="STYLE12">修改
        会员信息</span></td>
      </tr>
      <%
        GregorianCalendar d=new GregorianCalendar();
        String year=Integer.toString(d.get(Calendar.YEAR));
        String month=Integer.toString(d.get(Calendar.MONTH)+1);
        String day=Integer.toString(d.get(Calendar.DAY_OF_MONTH));
      %>
      <tr>
        <td height="40" class="STYLE2"><span class="STYLE10"><br />
          </span><span class="STYLE12">时间:<%=year%>-<%=month%>-<%=
          day%></span></td>
      </tr>
      <tr>
        <td height="30" align="center" class="STYLE12"><a href="cartmanage.jsp">返回会
        员首页</a></td>
```

```
    </tr>
  </table>
 </body>
<!-- InstanceEnd --></html>
```

【orderdetail.jsp】

```
<%@ page contentType="text/html; charset=gb2312" language="java"
import="java.sql.*,java.util.*" errorPage="" %>
<%@ page import="bean.cart"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.
w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"><!-- InstanceBegin template=
"/Templates/cart.dwt.jsp" codeOutsideHTMLIsLocked="false" -->
 <head>
  <meta http-equiv="Content-Type" content="text/html; charset=gb2312" />
  <!-- InstanceBeginEditable name="doctitle" -->
  <title>无标题文档</title>
  <!-- InstanceEndEditable -->
  <!-- InstanceBeginEditable name="head" -->
  <style type="text/css">
   <!--
    .STYLE19 {font-size: 18}
   -->
  </style>
  <!-- InstanceEndEditable -->
  <style type="text/css">
   <!--
    .STYLE14 {color: #FF6600}
    .STYLE18 {font-size: 24}
   -->
  </style>
  <style type="text/css">
   <!--
    .STYLE10 {font-family: "宋体"}
    .STYLE12 {font-family: Arial, Helvetica, sans-serif; font-size: 16px; }
    .STYLE2 {font-family: "华文行楷";
      font-size: 24px;
      color: #FF9900;
      }
    body {
       margin-left: 0px;
       margin-top: 0px;
       margin-right: 0px;
       margin-bottom: 0px;
       }
   -->
  </style>
</head>
<jsp:useBean id="db" class="bean.mydb" scope="page"/>
<body>
<table width="900" border="0" align="center" cellpadding="0" cellspacing="0">
```

```
<tr>
 <td width="149" height="130"><span class="STYLE2"><%=session.getAttribute("
 username")%>欢迎光临本网站选购图书</span></td>
 <td width="751" rowspan="7" align="center" valign="top"><!-- InstanceBeginEditable
 name="EditRegion1" -->
   <table width="700" border="0" cellspacing="0" cellpadding="0">
     <tr>
        <td height="30" align="center" valign="middle"><span class="STYLE10
        STYLE14">订单号</span></td>
        <td height="30" align="center" valign="middle"><span class="STYLE10
        STYLE14">用户名</span></td>
        <td height="30" align="center" valign="middle"><span class="STYLE10
        STYLE14">图书名称</span></td>
        <td height="30" align="center" valign="middle"><span class="STYLE10
        STYLE14">数量</span></td>
        <td height="30" align="center" valign="middle"><span class="STYLE10
        STYLE14">单价</span></td>
        <td height="30" align="center" valign="middle"><span class="STYLE10
        STYLE14">总额(元)</span></td>
     </tr>
     <%
       String id=request.getParameter("id");
       //float money=0;
       db.connection();
       String sql="SELECT * FROM orderlist WHERE id='"+id+"'";
       ResultSet rs=db.query(sql);
        rs.next();
       String s1=rs.getString(1);                 //订单号
       String s2=rs.getString(2);                 //用户名
       //String s3=rs.getString(3);               //图书名称
       //String s4=rs.getString(4);               //数量
       String s5=rs.getString(5);                 //总额
       //String s6=rs.getString(6);               //单价
       Vector list=(Vector)session.getAttribute("cartList");
       int n=list.size();
       //out.println(n);
        for(int i=0;i<n;i++)
          {
           cart cart=(cart)list.elementAt(i);
           sql="SELECT name,price "+"FROM book WHERE id='"+cart.getBookId()+"'";
           db.connection();
           rs=db.query(sql);
           rs.next();
           String s3=rs.getString(1);             //图书名称
           String s4=rs.getString(2);             //单价
           //out.println(sql);
     %>
     <tr>
       <td height="30" align="center" valign="middle"> <%=s1 %></td>
```

```
            <td height="30" align="center" valign="middle"> <%=s2 %></td>
            <td height="30" align="center" valign="middle"> <%=s3 %></td>
            <td height="30" align="center" valign="middle"> <%=cart.getNum()%>
            </td>
            <td height="30" align="center" valign="middle"> <%=s4 %></td>
            <td height="30" align="center" valign="middle"> <%=s5 %></td>
          </tr>
          <%} %>
        </table>
      <!-- InstanceEndEditable --></td>
    </tr>
    <tr>
      <td height="50" align="center" bgcolor="#FFCC66"><a href="cartmanage.jsp">
      <span class="STYLE12">购书</span></a></td>
    </tr>
    <tr>
      <td height="50" align="center" bgcolor="#FFCC66"><a href="mycart.jsp"><span
      class="STYLE12">我的购物车</span></a></td>
    </tr>
    <tr>
      <td height="50" align="center" bgcolor="#FFCC66"><a href="orderdisplay.jsp">
      <span class="STYLE12">查看订单</span></a></td>
    </tr>
    <tr>
      <td height="50" align="center" bgcolor="#FFCC66"><span class="STYLE12">修改
      会员信息</span></td>
    </tr>
    <%
      GregorianCalendar d=new GregorianCalendar();
      String year=Integer.toString(d.get(Calendar.YEAR));
      String month=Integer.toString(d.get(Calendar.MONTH)+1);
      String day=Integer.toString(d.get(Calendar.DAY_OF_MONTH));
    %>
    <tr>
      <td height="40" class="STYLE2"><span class="STYLE10"><br />
        </span><span class="STYLE12">时间:<%=year%>-<%=month%>-<%=
        day%></span></td>
    </tr>
    <tr>
      <td height="30" align="center" class="STYLE12"><a href="cartmanage.jsp">返回会
      员首页</a></td>
    </tr>
  </table>
</body>
<!-- InstanceEnd --></html>
```

【cartclear.jsp】

```
<%@ page contentType="text/html; charset=gb2312" language="java" import="java.sql.*"
errorPage="" %>
```

```
<%@page import="java.util.*"%>

<html>
 <body>
  <%
    Vector list=(Vector)session.getAttribute("cartList");
    list.removeAllElements();
    response.sendRedirect("mycart.jsp");
  %>
 </body>
</html>
```

【cartdelete.jsp】

```
<%@page contentType="text/html; charset=gb2312" language="java" import="java.sql.*" %>
<%@page import="java.util.*"%>
<%@page import="bean.cart"%>
<html>
 <jsp:useBean id="db" class="bean.mydb" scope="page"/>
 <body>
  <%
    String bookid = request.getParameter("id");
    Vector list = (Vector)session.getAttribute("cartList");
        if (list==null)
        {
            System.out.println("Error!");
        }
       //删除购买图书
       db.connection();
        for (int i=0; i<list.size(); i++)
        {
            cart icart= (cart)list.elementAt(i);
            if (bookid.equals(icart.getBookId()))
               {
                  list.removeElementAt(i);
                  break;
               } //if name matches 结束

        } // for 循环结束
    response.sendRedirect("mycart.jsp");
  %>
 </body>
</html>
```

附录 2

知 识 字 典

Appendtx 2

page 指令

page 指令中还有一些上面题目中没有用到的属性也为大家做一下介绍。

1. extends 属性

该属性用于指明 Java 类文件超类全名。使用该属性时要小心，因为，它会限制 JSP 引擎编译文件的能力。用法：extends="package. class"。

2. session 属性

该属性用于确定 JSP 页面是否使用 HTTP session。假如这个值是 true，则 session 引用的是当前或新的 session。假如这个值是 false，则在 JSP 文件中，不能使用 session 对象。默认值是 true。用法：session="true|false"。

3. buffer 属性

设置 buffer 缓冲区的大小，当 out 对象将被编译的 JSP 页面向客户 Web 浏览器端输出时使用。它的默认值是 8kb。假如你指定了 buffer 缓冲区的大小，这个输出量将不低于你指定的大小。用法：buffer="none|8kb|sizekb"。

4. autoFlush 属性

该属性指出：当缓冲区充满时，是否自动刷新。如果值为 true(默认值为 true)，则自动刷新；如果值为 false，则当缓冲区溢出时，将会产生错误。而且如果 buffer 属性的值设为 none，autoFlush 属性的值不能设为 false。用法：autoFlush="true|false"。

5. isThreadSafe 属性

假设这个值设为 true，多个线程能同时访问一个 JSP 页面；假如这个值为 false，对一个 JSP 页面，JSP 引擎一次只能响应一个用户的请求。默认值为 true。用法：isThreadSafe="true|false"。

6. info 属性

关于作者、版本和著作权等信息，可以通过 javax. sevlet. Servlet. getServletInfo()方法查到该信息的详细情况。用法：info="text"。

7. errorPage 属性

该属性用于设置处理例外事件的 JSP 文件的路径名。假如这个路径名以“/”开始，则这个路径名是对 JSP 应用文档根目录而言的，而且由 Web 服务器决定。否则，这个路

径是相对于当前 JSP 文件而言的。用法：errorPage="relativeURL"。

8. isErrorPage 属性

JSP 文件是否显示错误页面。如果这个值为 true，可以使用例外 exception 对象；如果这个值为 false(默认值也为 false)，在 JSP 文件中，不能使用 exception 对象。用法：isErrorPage="true|false"。

Date 对象的方法

Date 对象的方法如附表 2-1 所示。

附表 2-1 Date 对象的方法

获取日期和时间的方法	说明	设置日期和时间的方法	说明
getFullYear()	返回用 4 位数表示的年份	setFullYear()	设置年份，用 4 位数表示
getMonth()	返回月份(0～11)	setMonth()	设置月份(0～11)
getDate()	返回日数(1～31)	setDate()	设置日数(1～31)
getDay()	返回星期(0～6)	setDay()	设置星期(0～6)
getHours()	返回小时数(0～23)	setHours()	设置小时数(0～23)
getMinutes()	返回分钟数(0～59)	setMinutes()	设置分钟数(0～59)
getSeconds()	返回秒数(0～59)	setSeconds()	设置秒数(0～59)
getTime()	返回 Date 对象的内部毫秒表示	setTime()	使用毫秒形式设置 Date 对象

<input>标记

写入标记格式如下：

<input type=参数 name=参数 …>

type 属性值如附表 2-2 所示。

附表 2-2 type 属性值列表

属性	输入区域类型
text	单行的文本输入区域，size 与 maxlength 属性用来定义此种输入区域显示的尺寸大小与输入的最大字符数
button	普通按钮，当这个按钮被点击时，就会调用属性 onclick 指定的函数；在使用这个按钮时，一般配合使用 value 指定在它上面显示的文字，用 onclick 指定一个函数，一般为 JavaScript 的一个事件
submit	提交到服务器的按钮，当这个按钮被点击时，就会连接到表单 form 属性 action 指定的 url 地址
reset	重置按钮，单击该按钮可将表单内容全部清除，重新输入数据
checkbox	一个复选框，checked 属性用来设置该复选框缺省时是否被选中，右边示例中使用了三个复选框
hidden	隐藏区域，用户不能在其中输入，用来预设某些要传送的信息

续表

属　性	输入区域类型
image	使用图像来代替 Submit 按钮，图像的源文件名由 src 属性指定，用户点击后，表单中的信息和点击位置的 X、Y 坐标一起传送给服务器
password	输入密码的区域，当用户输入密码时，区域内将会显示"*"号
radio	单选按钮类型，checked 属性用来设置该单选框缺省时是否被选中，右边示例中使用了三个单选框

<jsp:include>动作标识与 include 指令之间存在的差异

<jsp:include>动作标识与 include 指令都可用来包含文件，下面来介绍它们之间存在的差异。

1. 属性

include 指令通过 file 属性来指定被包含的页面，include 指令将 file 属性值看作一个实际存在的文件的路径，所以该属性不支持任何表达式。若在 file 属性值中应用 JSP 表达式，则会抛出异常。

<jsp:include>动作标识通过 page 属性来指定被包含的页面，该属性支持 JSP 表达式。

2. 处理方式

使用 include 指令被包含的文件，它的内容会原封不动插入到包含页中使用该命令的位置，然后 JSP 编译器再对这个合成的文件进行编译。所以在一个 JSP 页面中使用 include 指令来包含另一个 JSP 页面，最终编译后的文件只有一个。

使用<jsp:include>动作标识包含文件时，当该标识被执行时，程序会将请求转发到（注意是转发，而不是请求重定向）被包含的页面，并将执行结果输出到浏览器中，然后返回包含页继续执行后面的代码。因为服务器执行的是两个文件，所以 JSP 编译器会分别对这两个文件进行编译。

3. 包含方式不同

上面讲到使用 include 指令包含文件，最终服务器执行的是将两个文件合成后由 JSP 编译器编译成的一个 Class 文件，所以被包含文件的内容应该是固定不变的，若改变了被包含的文件，则主文件的代码就发生了改变，因此服务器会重新编译主文件。include 指令的这种包含过程称为静态包含。

使用<jsp:include>动作标识通常是来包含那些经常需要改动的文件。此时服务器执行的是两个文件，被包含文件的改动不会影响到主文件，因此服务器不会对主文件重新编译，而只需重新编译被包含的文件即可。而对被包含文件的编译是在执行时才进行的，也就是说，只有当<jsp:include>动作标识被执行时，使用该识包含的目标文件才会被编译，否则被包含的文件不会被编译，所以这种包含过程称为动态包含。

out 对象的常用方法列举如下

1. clear()

此方法清除缓冲区中的数据，并不输出。调用方法如下：

```
out.clear()
```

2. clearBuffer()

此方法清除缓冲区中的数据，并把清除的缓冲区中的数据输出到客户端。调用方法如下：

```
out.clearBuffer()
```

3. close()

此方法用来关闭输出流。因为 out 对象实际上是一个输出流。调用方法如下：

```
out.close()
```

4. flush()

此方法输出缓冲区中的数据内容。调用方法如下：

```
out.flush()
```

5. getBufferSize()

此方法得到缓冲区的大小。调用方法如下：

```
out.getBufferSize()
```

6. getRemaining()

此方法得到缓冲区中未被使用空间的大小。调用方法如下：

```
out.getRemaining()
```

7. isAutoFlush()

此方法是否是自动刷新，返回的是一个布尔值。如果是自动刷新，返回 true，否则返回 false。调用方法如下：

```
out.isAutoFlush()
```

8. newLine()

此方法输出一个换行字符。调用方法如下：

```
out.newLine()
```

9. print()与 println()

print()与 println()方法应用广泛，可以输出各种类型的数据。

request 对象

request 的方法主要用来处理客户端浏览器提交请求中的各项参数和选项。主要方法有如下几种。

1. getAttribute()

此方法用来返回指定的属性值，如果不存在这个属性，返回 null 值。调用方法如下：

```
request.getAttribute(String str)
```

其中,参数 str 是指定的属性的名称。

2. getAttributeNames()

此方法返回 request 对象所有属性的名称,其返回值是一个枚举类型的对象。调用方法如下:

```
request.getAttributeNames()
```

3. getContentLength()

此方法获取客户端提交的所有信息的总长度。调用方法如下:

```
request.getContentLength()
```

4. getCookie()

此方法获得客户端的 Cookies 对象,其返回值为一个 Cookies 数组。调用方法如下:

```
request.getCookie()
```

5. getHeader()

此方法获取 HTTP 中指定的头名称值。调用方法如下:

```
request.getHeader(String str)
```

6. getHeaders()

此方法返回指定的头名称的所有值,返回值是一个枚举类型的对象。调用方法如下:

```
request.getHeaders(String str)
```

其中,参数 str 是指定的头名称。

7. getHeadersName()

此方法获得 request 的头中属性的名称,其返回值是一个枚举类型的对象。调用方法如下:

```
request.getHeadersName()
```

8. getMethod()

此方法获得 request 方法,如 get 或 post 等。调用方法如下:

```
request.getMethod()
```

9. getParameter()

此方法获得指定名称的客户端提交的参数值。调用方法如下:

```
request.getParameter(String str)
```

其中,参数 str 是指定的参数名称,这个方法是 JSP 程序设计中最为常用的方法之一。

10. getParameterNames()

此方法获得客户端提交的参数名称,其返回值是一个枚举类型对象。调用方法如下:

```
request.getParameterNames()
```

11. getParameterValues()

此方法获得指定客户端提交的参数值。调用方法如下：

```
request.getParameterValues(String str)
```

12. getProtocol()

此方法得到客户端浏览器向服务器端发送数据时所依据的协议名称。调用方法如下：

```
request.getProtocol()
```

13. getQueryString()

此方法得到查询字符串，这个字符串是客户端以 get 进行传递，或是在网页地址后加上参数方法得到的。调用方法如下：

```
request.getQueryString()
```

14. getRequestURI()

此方法获得发送请求字符串的客户端地址。调用方法如下：

```
request.getRequestURI()
```

15. getRemoteAddr()

此方法获得客户端的 IP 地址。调用方法如下：

```
request.getRemoteAddr()
```

16. getRemoteHost()

此方法获得客户端的机器名称。如果得不到名称则获得客户端的 IP 地址。调用方法如下：

```
request.getRemoteHost()
```

17. getServerName()

此方法获得服务器的机器名称。调用方法如下：

```
request.getServerName()
```

18. getServletPath()

此方法返回客户端请求页面的文件目录。调用方法如下：

```
request.getServletPath()
```

19. getServerPort()

此方法获得服务器端提供 HTTP 服务的端口号。调用方法如下：

```
request.getServerPort()
```

Session 对象的方法

1. getAttribute()

此方法获得指定的 session 对象。调用方法如下：

```
session.getAttribute(String name)
```

其中，参数 name 是指定 session 对象的名称。该方法返回一个 Object 对象，任何对象都可加入到 session 对象中，因此对返回的对象要用强制转换的方法，将得到的对象转换为此对象原来的类型。

2. getAttributeNames()

此方法得到 session 对象中所有数据名称的枚举。调用方法如下：

```
session.getAttributeNames()
```

该方法返回一个枚举类型的对象，可以用返回对象的 nextElements()方法来遍历 session 对象中所有的数据对象。

3. getCrationTime()

此方法得到 session 对象创建的时间。调用方法如下：

```
session.getCreationTime()
```

此方法返回一个 long 类型的值，单位为毫秒，是从 1970 年 7 月 1 日午夜起至 session 对象创建为止所相隔的毫秒数。

4. getId()

此方法得到 session 对象的 ID。调用方法如下：

```
session.getId()
```

由于每个客户端与服务器对话的 ID 不同，因此服务器通过这个 ID 来区分不同的用户。

5. getMaxInactiveIterval()

此方法获得 session 对象的生存时间，返回一个 int 型值，单位是秒。调用方法如下：

```
session.getMaxInactiveIterval()
```

6. getLastAccessedTime()

此方法得到此处 session 对象最后一次被用户操作的时间，单位为毫秒，是从 1970 年 7 月 1 日午夜起至 session 对象最后一次被操作停止所相隔的毫秒数。调用方法如下：

```
session.getLastAccessedTime()
```

7. Invalidate()

此消息使 session 对象失效。调用方法如下：

```
Session.Invalidate()
```

8. isNew()

此方法判断是否是一个新的用户。调用方法如下:

```
session.isNew()
```

该方法返回一个布尔值,返回 true 表示一个新的用户。

9. removeAttribute()

这个方法用来从 session 对象中删除指定的 session 对象。调用方法如下:

```
session.removeAttribute(String name)
```

其参数 name 是指定的 session 对象的名称,该方法无法返回值。

10. setAttribute()

此方法用来设置指定 session 对象中数据对象的值。调用方法如下:

```
session.setAttribute(String name,object value)
```

其中,第一个参数 name 指出要设置的 session 对象中数据对象的名称;第二个参数 value 使这个对象的对应值。如果指定的对象不存在,则新加入一个;如果已存在,则该值会被更改。

11. SetMaxInactiveInterval()

此方法设置 session 对象生产的时间。调用方法如下:

```
session.SetMaxInactiveInterval(int timecount)
```

其中,参数 timecount 指出生存的时间,单位是秒,该方法无返回值。

application 对象常用方法

1. getAttribute()

此方法得到指定的 application 对象在数据对象的值。调用方法如下:

```
application.getAttractive(string name)
```

2. getAttributeNames()

此方法得到 application 对象所有数据对象的一个枚举。调用方法如下:

```
application.getAttributeNames()
```

这个方法返回一个枚举类型的对象,可以用返回对象的 nextElements 方法来遍历 application 对象中所有的数据对象。

3. getInitParameter()

此方法返回指定的 application 对象中数据对象的初始值。调用方法如下:

```
application.getInitParameter()
```

其中,参数 name 是指定 application 对象中数据对象的名称。

4. getServletInfo()

此方法返回 Servlet 编译器当前的版本信息。调用方法如下：

```
application.getServletInfo()
```

5. setAttribute()

此方法设置指定名称的 application 对象的属性值。调用方法如下：

```
application.setAttribute()
```

其中，第一个对象 name 是指 application 对象中数据对象的名称：第二个参数 object 是这个对象对应的值，其类型是 Object，说明可以是任意类型，如果对象不存在，则添加该对象如果已存在则将其覆盖。

附录 3

习 题 集 锦

Appendtx 3

习 题 一

一、选择题

1. JSP 的编译指令标记通常是指(　　)。

   A. Pag 指令、Include 指令和 Taglib 指令

   B. Pag 指令、Include 指令和 Plugin 指令

   C. Forward 指令、Include 指令和 Taglib 指令

   D. Pag 指令、Include 指令和 Taglib 指令

2. 可以在(　　)标记之间插入 Java 程序片。

   A. <%和%>　　B. <%和/>　　C. </和%>　　D. <%和! >

3. (　　)不属于 JSP 动作指令标记。

   A. < jsp: param >　　B. < jsp:plugin >

   C. < jsp:useBean >　　D. < jsp:javaBean >

4. JSP 的 Page 编译指令的属性 Language 的默认值是(　　)。

   A. Java　　B. C　　C. C#　　D. SQL

5. JSP 的(　　)指令允许页面使用者自定义标签库。

   A. Include　　B. Taglib　　C. Page　　D. Plugin

6. 可以在(　　)标记之间插入变量与方法声明。

   A. <%和%>　　B. <%和/>　　C. </和%>　　D. <%和! >

7. 能够替代<字符的字符是(　　)。

   A. &lt　　B. &gt　　C. &lt　　D.

8. 下列(　　)注释为隐藏性注释。

   A. <! - -注释内容[<% =表达式%>] - - >

   B. <! - -注释内容- - >

   C. <% - -注释内容- - %>

   D. <! - - [<% =表达式% >] - - >

9．下列变量声明在(　　)范围内有效。

```
<%! Date dateTime;
    Int eountNum;
%>
```

A．从定义开始处有效，客户之间不共享

B．在整个页面内有效，客户之间不共享

C．在整个页面内有效，被多个客户共享

D．从定义开始处有效，被多个客户共享

10．在<%！和%>标记之间声明的 Java 的方法称为页面的成员方法，其在(　　)范围内有效。

A．从定义处之后有效　　B．在整个页面内有效

C．从定义处之前有效　　D．不确定

11．在<%！和%>标记之间放置(　　)，可以直接输出其值。

A．变量　　B．Java 表达式　　C．字符串　　D．数字

12．include 指令用于在 JSP 页面静态插入一个文件，插入文件可以是 JSP 页面、HTML 页面、文本文件或一段 Java 代码，但必须保证插入后形成的文件是(　　)。

A．是一个完整的 HTML 文件　　B．是一个完整的 JSP 文件

C．是一个完整的 TXT 文件　　D．是一个完整的 Java 源文件

13．当一个客户线程执行某个方法时，其他客户必须等待，直到这个客户线程调用执行完毕该方法后，其他客户线程才能执行，这样的方法在定义时必须使用关键字(　　)。

A．public　　B．static　　C．synchronized　　D．private

二、判断题

1．在 HTML 页面文件中加入 JSP 脚本元素、JSP 标记等就构成了一个 JSP 页面。(　　)

2．JSP 页面中的变量和方法声明(declaration)、表达式(Expression)和 Java 程序片段(Scriptlet)统称为 JSP 标记。(　　)

3．JSP 页面中的指令标记、JSP 动作标记统称为脚本元素。(　　)

4．在<%！和%>标记之间声明的 Java 变量在整个页面内有效，不同的客户之间不共享。(　　)

5．在<%！和%>标记之间声明的 Java 变量在整个页面内有效。(　　)

6．页面成员方法不可以在页面的 Java 程序片中调用。(　　)

7．程序片变量不同于在"<%!"和"%>"之间声明的页面成员变量，不能在不同客户访问页面的线程之间共享。(　　)

8．JSP 中 Java 表达式的值由服务器负责计算，并将计算值按字符串发送给客户端显示。(　　)

9．在 Java 程序片中可以使用 Java 语言的注释方法，其注释的内容会发送到客户端。(　　)

10．不可以用一个 page 指令指定多个属性的取值。(　　)

11. jsp：include 动作标记与 include 指令标记包含文件的处理时间和方式不同。 ( )

12. jsp：param 动作标记不能单独使用，必须作为 jsp：include、jsp：forward 标记等的子标记使用，并为它们提供参数。 ( )

13. <jsp：forward..>标记的 page 属性值是相对的 URL 地址，只能静态的 URL。 ( )

14. JSP 页面中不能包含脚本元素。 ( )

15. Page 指令不能定义当前 JSP 程序的全局属性。 ( )

三、填空题

1. 一个完整的 JSP 页面是由普通的 HTML 标记、JSP 指令标记、JSP 动作标记、变量声明与方法声明、________、________、________ 7 种要素构成。

2. JSP 页面的基本构成元素，其中变量和方法声明（Declaration）、表达式（Expression）和 Java 程序片（Scriptlet）统称为________。

3. 指令标记、JSP 动作标记统称为________。

4. JSP 页面的程序片中可以插入________标记。

5. 当 JSP 页面的一个客户线程在执行________方法时，其他客户必须等待。

6. JSP 页面中，输出型注释的内容写在________和________之间。

7. Page 指令的属性 Language 的默认值是________。

四、思考题

1. include 标记与 include 动作标记有什么区别？

2. 如何保证页面跳转时当前页面与跳转页面之间的联系？

3. 如果有两个用户访问一个 JSP 页面，该页面的程序片将被执行几次？

4. 在<%！和%>之间声明的变量和在<%和%>之间声明的变量有何区别？

5. 是否允许一个 JSP 页面为 contentType 设置两次不同的值？

6. JSP 的特殊字符与 Java 语言的转义字符有何关系？

7. 叙述一个 JSP 页面的基本组成。

习 题 二

一、选择题

1. 下面不属于 JSP 内置对象的是（ ）。

A. out B. respone C. application D. page

2. 以下（ ）对象提供了访问和放置页面中共享数据的方式。

A. pageContext B. response C. request D. session

3. 调用 getCreationTime（ ）可以获取 session 对象创建的时间，该时间的单位是（ ）。

A. 秒 B. 分秒 C. 毫秒 D. 微秒

4. 一个典型的 HTTP 请求消息包括请求行、多个请求和（ ）。

A. 响应行 B. 信息体 C. 响应行 D. 响应头

5. out 对象是一个输出流，其输出各种类型数据并换行的方法是(　　)。

A. out. print()　　B. out. newLine()　　C. out. println　　D. out. write()

6. out 对象是一个输出流，其输出不换行的方法是(　　)。

A. out. print()　　B. out. newLine()　　C. out. println　　D. out. write()

7. Form 表单的 method 属性能取下列(　　)的值。

A. submit　　B. puts　　C. post　　D. out

8. 能在浏览器的地址栏中看到提交数据的表单提示方式是(　　)。

A. submit　　B. get　　C. post　　D. out

9. 可以利用 request 对象的(　　)方法获取客户端的表单信息。

A. request. getParameter()　　B. request. outParameter()

C. request. writeParameter()　　D. request. handlerParameter()

10. JSP 页面中 request. getParamter()得到的数据，其类型是(　　)。

A. Double　　B. int　　C. String　　D. Integer

11. JSP 页面程序片中可以使用(　　)方法将 strNumx= request. getParamter("ix")得到的数据类型转换为 Double 类型。

A. Double. parseString(strNumx)　　B. Double. parseDpuble(strNumx)

C. Double. parseInteger(strNumx)　　D. Double. parseFloat(strNumx)

12. 下面不属于<input>标记中的 name 属性取值的是(　　)。

A. text　　B. radio　　C. checkbox　　D. picture

13. <select>用于在表单中来定义下列表框和滚动列表框空间，下面(　　)属性指定列表框默认选项。

A. size　　B. value　　C. selected　　D. checked

二、判断题

1. Tomcat 服务器支持直接使用 application 对象。(　　)

2. out 对象是一个输出流，它实现了 javax. servlet. JspWriter 接口，用来向客户端输出数据。(　　)

3. 利用 response 对象的 sendRedirect 方法只能实现本网站内的页面跳转，但不能传递参数。(　　)

4. respone 对象主要用于向客户端发送数据。(　　)

5. contentType 属性用来设置 JSP 页面的 MIME 类型和字符编码集，取值格式为"MIME 类型"或"MIME 类型；charset ＝字符编码集"，response 对象调用 addHeader 方法修改该属性的值。(　　)

6. Post 属于表单的隐式提交信息方法。(　　)

7. < select >标记用于在表单中插入一个下拉菜单。(　　)

8. 表单信息的验证只能放在服务器端执行。(　　)

9. 网页中只要使用 GB2312 编码就不会出现中文乱码。(　　)

10. 表单提交的信息就封装在 HTTP 请求消息的信息体部分，用户使用 request 对象的 getParameter 方法可以得到通过表单提交的信息。(　　)

11. request 对象的 getRemoteHost()方法既能获取客户名称,又能获取客户 IP 地址。 (　　)

12. 同一个客户在同一个 Web 服务目录中的 session 对象是相通的,在不同的 Web 服务器目录中的 session 对象不相同。 (　　)

13. session 对象是 HttpSession 接口类的实例,由客户端负责创建和销毁,所以不同客户的 session 对象不同。 (　　)

14. public long session. setMaxInactiveInterval()设置最长发呆时间,单位毫秒。 (　　)

15. session 对象可以用来保存用户会话期间需要保存的数据信息。 (　　)

16. application 对象对所有用户都是共享的,任何对它的操作都会影响到所有的用户。 (　　)

三、填空题

1. out 对象的________方法,功能是输出缓冲的内容。
2. JSP 的________对象用来保存单个用户访问时的一些信息。
3. response 对象的________方法可以将当前客户端的请求转到其他页面去。
4. 当客户端请求一个 JSP 页面时,JSP 容器会将请求信息包装在________对象中。
5. response. setHeader("Rsfresh","5")的含义是指页面刷新时间为________。
6. 表单的提交方法包括________和________方法。
7. 通过设置 Cookie 变量的________属性来设置他的生命期限。

四、思考题

1. 请简述 JSP 中常用的内置对象。
2. 简述 request 对象和 response 对象的作用。
3. session 对象与 application 对象有何区别?
4. 网页中的表单如何定义,通常表单中包含哪些元素?
5. 如何处理表单提交的汉字?
6. 一个用户在不同的 Web 服务目录的 session 相同吗?
7. 内置对象的 4 个作用范围是什么? 什么情况下 session 会关闭?
8. response. sendRedirect(URL url)方法的作用是什么?
9. 怎样使用 request、session 和 application 对象进行参数存取?
10. 什么是 Cookie? 怎么设置和获取 Cookie 值?

习　题　三

一、选择题

1. 下面(　　)不是 JDBC 的工作任务。
   A. 与数据库建立连接
   B. 操作数据库,处理数据库返回的结果
   C. 在网页中生成表格
   D. 向数据库管理系统发送 SQL 语句

2. 下面(　　)不是加载驱动程序的方法。

A. 通过 DriverManager. getConnection 方法加载

B. 调用方法 Class. forName

C. 通过添加系统的 jdbc. drivers 属性

D. 通过 registerDriver 方法

3. 关于分页显示,下列的叙述(　　)是不正确的。

A. 只编制一个页面是不可能实现分页显示的

B. 采用一个至三个页面都可以实现分页显示

C. 分页显示中,记录集不必在页面跳转后重新生成

D. 分页显示中页面显示的记录数可以随用户输入调整

4. DriverManager 类的 getConnection(String url, String user, String password)方法中,参数 url 的格式为 jdbc:＜子协议＞:＜子名称＞,(　　)url 是不正确的。

A. “jdbc: mysql://localhost:80/数据库名”

B. “jdbc:odbc:数据源”

C. “jdbc:oracle:thin@host:端口号:数据库名”

D. “jdbc:sqlserver://172.0.0.1:1443; DatabaseName =数据库名”

5. 在 JDBC 中,(　　)接口不能被 Connection 创建。

A. Statement　　B. PreparedStatement

C. CallableStatement　　D. RowsetStatement

6. 下面是加载 JDBC 数据库驱动的代码片段:

```
Try {
        Class. forName("sun.jdbc.odbc.JdbcOdbcDriver");
}
Catch(ClassNotFoundException e) {
Out.print (e);
}
```

该程序加载的是(　　)驱动。

A. JDBC-ODBC 桥连接驱动　　B. 部分 Java 编写本地驱动

C. 本地协议纯 Java 驱动　　D. 网络纯 Java 驱动

7. 下面是创建 Statement 接口并执行 executeUpadate 方法的代码片段:

```
Conn = DriverManager. getConnection ("jdbc:odbc:book",",");
Stmt= conn.createStatement();
String strsql="insert into book values("TP003', 'ASP.NET', '李', '清华出版社',35)";
n = stmt.executeUpdate(strsql);
```

代码执行成功后 n 的值为(　　)。

A. 1　　B. 0　　C. −1　　D. 一个整数

8. 下列代码中 rs 为查询得到的结果集,代码运行后表格的每一行有(　　)个单元格。

```
while (rs.next( )){
```

```
out.print ("<tr>");
        out.print("<td>"+ rs.getString(1) +"</td>");
        out.print("<td>"+ rs.getString(2) +"</td>");
        out.print("<td>"+ rs.getString(3) +"</td>");
        out.print("<td>"+ rs.getString("publish")+"</td>");
        out.print("<td>"+ rs.getFloat("price")+"</td>");
out.print("</tr>");
}
```

A. 4　　B. 5　　C. 6　　D. 不确定

9. 查询结果集 ResultSet 对象是以统一的行列形式组织数据的，执行 ResultSet rs =stmt.executeQuery("select bid , name , author ,publish ,price from book");语句，得到的结果集 rs 列数为(　　)。

A. 4　　B. 5　　C. 6　　D. 不确定

10. 下列代码生成了一个结果集

```
Conn = DriverManager.getConnection (uri , user , password);
Stmt = conn.createStatement (ResultSet.TPYE_SCROLL_SENSITIVE,
ResultSet.CONCUR_ READ_ ONLY);
rs=stmt.executeuery("select * from book");
```

下面对 rs 描述正确的是(　　)。

A. 只能向下移动的结果集　　B. 可上下滚动的结果集

C. 只能向上移动的结果集　　D. 不确定是否可以滚动

11. 给出了如下的查询条件字符串 String condition =" insert book values (?,?,?,?,?)";下列哪个接口适合执行该 SQL 查询？(　　)

A. Statement　　B. PrepareStatement

C. CallableStatement　　D. 不确定

二、判断题

1. JDBC 构建在 ODBC 基础上，为数据库应用开发人员、数据库前台工具开发人员提供了一种标准，使开发人员可以用任何语言编写完整的数据库应用程序。　(　　)

2. JDBC 加载不同数据库的驱动程序，使用相应的参数可以建立与各种数据库的连接。　(　　)

3. 应用程序分页显示记录集时，不宜在每页都重新连接和打开数据库。　(　　)

4. JDBC 中的 URI 提供了一种标识数据库的方法，使 DriverManage 类能够识别相应的驱动程序。　(　　)

5. 进行分页，可调用 JDBC 的规范中有关分页的接口。　(　　)

6. JDBC 的 URL 字符串是由驱动程序的编写者提供的，并非由该驱动程序的使用者指定。　(　　)

7. ResultSet 结果集，每一次 rs 可以看到一行，要在看到下一行，必须使用 next()方法移动当前行。　(　　)

8. 如果需要在结果集中前后移动或随机显示一条记录，这时就必须得到一个可以滚

动的结果集。（　　）

9. Statement 对象提供了 int executeUpdate (String sqlStatement)方法，用于实现对数据库中数据的添加、删除和更新操作。（　　）

10. Statement 对象的 executeUpdate(String sqlStatement)方法中，sqlStatement 参数是由 insert、delete 和 update 等关键字构成的 SQL 语句。（　　）

三、填空题

1. JDBC 的英文全称是________，中文意义是________。
2. 简单地说，JDBC 能够完成下列三件事：________、________、________。
3. JDBC 主要由两部分组成：一部分是访问数据库的高层接口，即通常所说的________；另一部分是由数据库厂商提供的使用 Java 程序能够与数据库连接通信的驱动程序，即________。
4. 目前 JDBC 驱动程序可以分为四类：________、________、________、________。
5. 数据库的连接是由 JDBC 的________管理的。
6. 查询结果集 RseultSet 对象是以统一的行列形式组织数据的，执行 RseultSet rs=stmt.executeQuery("select bid , name , author , publish , price from book");语句，得到结果集 rs 第一列对用________；而每一次 rs 只能看到________行，要再看到下一行，必须使用________方法移动当前行。ResultSet 对象使用方法获得当前行字段的值。
7. stmt 为 Statement 对象，执行 String sqlStatement ="delete from book where bid='tp1001'";语句后，删除数据库表的记录需要执行________语句。

四、思考题

1. 什么是 JDBC 编程接口？
2. 试列举一个自己在编程和调试中碰到的数据库问题及其解决办法。
3. 如何实现记录集的行操作？
4. 进行记录定位的方法有哪些？
5. 使用本地协议纯 Java 驱动操作 SQL Server 数据库的步骤？
6. 使用预处理语句和存储过程有什么好处？
7. 如何使用滚动的结果集？

习　题　四

一、选择题

1. JavaBean 可以通过相关 JSP 动作指令进行调用。下面(　　)不是 JavaBean 可以使用的 JSP 动作指令。

A. <jsp:useBean>　　B. <jsp:setProperty>

C. <jsp:getProperty>　　D. <jsp:setParameter>

2. 关于 JavaBean，(　　)叙述不正确。

A. JavaBean 的类必须是具体的和公共的，并且具有无参数的构造器

B. JavaBean 的类属性是私有的，要通过公共方法进行访问

C. JavaBean 和 Servlet 一样，使用之前必须在项目的 web.xml 中注册

D. JavaBean 属性和表单控件名称能很好地耦合，得到表单提交的参数

3. JavaBean 的属性必须声明为 private，方法必须声明为(　　)访问类型。

A. private　　B. static　　C. protect　　D. public

4. JSP 页面通过(　　)来识别 Bean 对象，可以在程序片中通过 xx. method 形式来调用 Bean 中的 set 和 get 方法。

A. name　　B. class　　C. id　　D. classname

5. JavaBean 的作用范围可以是 page、request、session 和(　　)四个作用范围中的一种。

A. application　　B. local　　C. global　　D. class

6. (　　)作用范围的 Bean()，请求响应完成则该 Bean 即被释放，不同客户的 Bean 互相不同。

A. application　　B. request　　C. page　　D. session

7. (　　)作用范围的 Bean()，被 Web 服务目录下所有用户共享，任何客户对 Bean 属性的修改都会影响到其他用户。

A. application　　B. request　　C. page　　D. session

8. (　　)作用范围的 Bean，当客户离开这个页面时，JSP 引擎取消为客户该页面分配的 Bean，释放他所占的内存空间。

A. application　　B. request　　C. page　　D. session

9. 使用＜jsp:getProperty＞动作标记可以在 JSP 页面中得到的 Bean 实例的属性值，并将其转换为什么类型的数据，发送到客户端(　　)。

A. String　　B. Double　　C. Object　　D. Classes

10. 使用＜jsp:setProperty＞动作标记可以在 JSP 页面中设置 Bean 的属性，但必须保证 Bean 有对应的(　　)方法。

A. Setxxx　　B. setxxx　　C. getxxx　　D. Getxxx

11. 使用格式＜jsp:setProperty name = "beanid" propery ="bean 的属性" value = "＜％ = expression ％＞"/＞给 Bean 的属性赋值，expression 的数据类型和 bean 的属性类型(　　)。

A. 必须一致　　B. 可以不一致　　C. 必须不同　　D. 无要求

12. 在 JSP 页面中使用＜jsp:setProperty name = “beanid” property =”bean 的属性” value =”字符串”/＞格式给 Long 类型的 Bean 属性赋值，会调用(　　)数据类型转换方法。

A. Long. parseLong(String s)　　B. Integer. parslnt (String s)

C. Double. parseDouble (String s)　　D. 不确定

13. (　　)调用数据类型转换方法会发生 NumberFormatException 一场。

A. Long. parseLong("1234")　　B. Integer. parseInt("1234")

C. Double. parseDouble("123. 45")　　D. Integer. parseInt("123a")

14. 在 JSP 页面中使用＜jsp:setProperty name ="bean 的名字" property =" * " /＞格式，将表单参数为 Bean 属性赋值，property = " * "格式要求 Bean 的属性名字(　　)。

A. 必须和表单参数类型一致　　B. 必须和表单参数名称一一对应

C. 必须和表单参数数量一致　　　　D. 名称不一定对应

15. 在 JSP 页面中使用<jsp:setPropety name ="bean 的名字" property ="bean 属性名" param = "表单参数名"/>格式,用表单参数为 Bean 属性赋值,要求 Bean 的属性名字(　　)。

A. 必须和表单参数类型一致　　　　B. 必须和表单参数名称一一对应

C. 必须和表单参数数量一致　　　　D. 名称不一定对应

二、判断题

1. JavaBean 的属性可读写,编写时 set 方法和 get 方法必须配对。　(　　)

2. JavaBean 也是 Java 类,因此也必须有主函数。　(　　)

3. JavaBean 组建就是 Java 开发中的一个类,通过封装属性和方法称为具有某种功能和接口的类,所以具有 Java 程序的特点。　(　　)

4. Sun 公司把 JavaBean 定义为一个可重复使用的软件组件,类似于电脑 CPU、硬盘等组件。　(　　)

5. JavaBean 分为可视化组建和非可视化组建。　(　　)

6. JavaBean 的属性必须声明为 private,方法必须声明为 public 访问类型。　(　　)

7. 创建 JavaBean 要经过编写代码、编译源文件、配置 JavaBean 这样一个过程。　(　　)

8. 在 JSP 页面中调用的 Bean 类中如果有构造方法,必须是 public 类型且必有参数。　(　　)

9. 布置 JavaBean 须在 Web 服务目录的 WEB-INF\ classes 子目录下建立与包名对应的子目录,并将字节文件复制到该目录中。　(　　)

10. JavaBean 中,对于 boolean 类型的属性,可以使用 is 代替方法名称中的 set 和 get 前缀,创建 Bean 必须带有包名。　(　　)

11. 在 JSP 页面中使用 Bean 首先要使用 import 指令将 Bean 引入。　(　　)

12. 修改了 Bean 的字节码后,要将新的字节码复制到对应的 WEB-INF \classes 目录中,重新启动 tomcat 服务器才能生效。　(　　)

13. 客户在某个页面修改 session 作用范围 Bean 的属性,在其他页面,该 Bean 的属性会发生同样的变化,不同客户之间的 Bean 也发生变化。　(　　)

14. 使用<jsp:setProperty>动作标记,可以使用表达式或字符串为 Bean 的属性赋值。　(　　)

15. 使用格式<jsp:setProperty name =" beanid" property =""bean 的属性" value="字符串"/>给 Bean 的属性赋值,这个字符串会自动被转化为属性的数据类型。　(　　)

16. 表单提交后,<jsp:setProperty>动作指令才会被执行。　(　　)

三、填空题

1. 在 Web 服务器端使用 JavaBean,将原来页面中程序片完成的功能封装到 JavaBean 中,这样能很好地实现________。

2. JavaBean 中用一组 set 方法设置 Bean 的私有属性值,get 方法获得 Bean 的私有属性值。set 和 get 方法名称与属性名称之间必须对应,也就是:如果属性名称为 xxx,那么 set 和 get 方的名称必须为________和________。

3. 用户在实际 Web 应用开发中，编写 Bean 除了要使用 import 语句引入 Java 的标准类外，可能还需要自己编写的其他类。用户自己编写的被 Bean 引用的类称为________。

4. 创建 JavaBean 的过程和编写 Java 类的过程基本相似，可以在任何 Java 的编程环境下完成________、________、________、________。

5. 布置 JavaBean 要在 Web 服务目录的 WEB-INF/classes 文件中建立与________对应的子目录，用户要注意目录名称的大小写。

6. 使用 Bean 首先要在 JSP 页面中使用________指令将 Bean 引入。

7. 要想在 JSP 页面中使用 Bean，必须首先使用________动作标记在页面中定义一个 JavaBean 的实例。

8. scope 属性代表了 JavaBean 的作用范围，它可以是 page、________、session 和 application 四个作用范围中的任何一种。

四、思考题

1. JavaBean 和一般意义上的 Java 类有何区别?

2. 简述 Bean 的编写方法，有哪些注意要点。

3. 如何实现一个 Bean 的属性与表单参数的关联?

4. 如何在页面的程序片中使用 Bean?

5. 试述 request、session 和 application 有效范围的 Bean 生命周期。

参 考 文 献

[1] 潘国荣. Java&JSP应用程序实例开发[M]. 北京：电子工业出版社，2014.
[2] 刘勇军，韩最蛟. Java Web核心编程技术(JSP、Servlet编程)[M]. 北京：电子工业出版社，2014.
[3] 刘素芳. JSP动态网站开发案例教程[M]. 北京：机械工业出版社，2014.
[4] 刘瑞新. JSP＋MySQL＋Dreamweaver动态网站开发实例教程[M]. 北京：机械工业出版社，2013.
[5] 耿祥义，张跃平. JSP大学实用教程(第2版)[M]. 北京：电子工业出版社，2013.
[6] 姜强，赵蔚. SQL Server与JSP动态网站开发——从设计思想到编程实战[M]. 北京：电子工业出版社，2013.
[7] 刘志成，宁云. JSP程序设计案例教程[M]. 北京：高等教育出版社，2013.
[8] 郑阿奇. JSP编程教程[M]. 北京：电子工业出版社，2012.
[9] 吴鹏. 动态网页设计(JSP)[M]. 北京：高等教育出版社，2011.
[10] 汪赵强. 网页制作与JSP技术(第2版)[M]. 北京：机械工业出版社，2010.
[11] 贾素玲，王强. JSP应用开发技术[M]. 北京：清华大学出版社，2007.
[12] 邓子云，燕峰. JSP网络编程从基础到实践(第2版)[M]. 北京：电子工业出版社，2007.
[13] 申吉红，廖学峰. JSP课程设计案例精编[M]. 北京：清华大学出版社，2007.
[14] 彭超，马丁. 新一代JSP网络编程入门与实践[M]. 北京：清华大学出版社，2007.
[15] 石志国. JSP网络开发详解[M]. 北京：电子工业出版社，2007.
[16] 邹竹彪. JSP网络编程从入门到精通[M]. 北京：清华大学出版社，2007.
[17] 王国辉，王毅，李钟尉. JSP程序开发范例宝典[M]. 北京：人民邮电出版社，2007.